مجلة

مجتمع لينوكس العربي

مجلة تعني بشؤون المصادر الحرة
العدد (٩) فبراير ٢٠١٠
www.linuxac.org



إقرأ أيضاً

أخبار خفيفة

لأنك فيدورا (شعر فصحى)

مدير الملفات ZYPPER

تقنية تبادل الملفات Magnet

مثلث أمن المعلومات
و المصادر المفتوحة

و مواضيع أخرى متعددة

مقدمة

بسم الله الرحمن الرحيم



نحمد الله ونستعين به ونتوكل عليه ونؤول إليه ونشكره تعالى قائلين
اللهم لا علم لنا إلا ما علمتنا إنك أنت العليم الحكيم.

أخوتي أعضاء مجتمع لينوكس العربي ، أحييكم بأفضل تحية ، تحية ملؤها سلام ورحمة وبركات ، تحية الرسل وتحية المؤمنين في الدنيا والآخرة ، كما يقول عز وجل "وتحيتهم فيها سلام" فسلام من الله عليكم ورحمته وبركاته أما بعد.

انطلقنا في بناء رسالتنا لعالمنا العربي والإسلامي ، عندما ركبنا بحار العلم ، بعد أن صنعنا سفينتنا بأنفسنا ولم نطلق عليها إسم الـ"تايتنك" ولا حتى "تشالنجر" ، بل أسميناها "مجتمع لينوكس العربي" ، وقد جعلنا مجاديفها أقلاماً متعطشة للعلم ومهووسة بالمنفعة العامة ، وجعلنا حمولتها علوماً ودروساً ينتفع بها وتدوم إلى نهاية الزمان بإذنه تعالى.

إن التحدي الذي نواجهه اليوم هو تناقص معدل تسارع سفينتنا هذه و التي نرتحل بها إلى بر الأمان، وكحال أي سفينة كيفما كان حالها فإنها لا تريد ريحا تجري بما لا تشتهي، وتلك الرياح التي تعاكس اتجاهنا ما هي إلا قليل من ظروف الحياة ومشاغلها الكثيرة ، وقليل آخر من تقاعس وتكاسل.

إن سفينتنا هذه تتميز عن غيرها من السفن، فلكل سفينة ربان واحد وبحارة كثر، ولكن ليس سفينتنا فجميعنا بحارة وربان، نحن جميعا نقودها ونوجهها وندعمها ونقويها ، نحن جميعا نعتني بها ونطور عملها ونحسن من أدائها، إن سفينتنا تبحر بأقلامنا وكتاباتنا ، ومداد قلمكم هو ما يحركها في بحر مظلم لا يضيئه سوى العقول المنيرة التي لطالما أنارت طريقها دون منارة على شاطئ تدلنا أي طريق نسلك.

إن الدور الملقى على عاتقنا اليوم هو إعادة إحياء العلوم والمقالات المتميزة التي لطالما كانت تتنافس فيما بينها على الارتقاء بعقول مرتادينا وتزويدهم بالمعرفة والعلم والحلول لمشاكلهم، كما أن دورنا يتمثل بنشر رسالتنا أكثر وأكثر، فجميعنا نمثل مجتمع لينوكس العربي، وجميعنا نريد له التقدم والتطور والسمو، فلا يخبو نوره ولا تنطفئ شعلته ، ولا يثنينا شيء عن ذلك.

فيا "ربابين" مجتمع لينوكس العربي، ضعوا "قبعاتكم" على رؤوسكم ، وتولوا توجيه "الدفة" ، أمامنا طريق طويل وبحر معتم ، لا تيأسوا ولا تضعفوا لأي عاصفة تواجهنا، فنحن لها وعلى قدر أهل العزم تأتي العزائم.

رئيس التحرير

سامر حداد

فهرس

أخبار خفيفة

بقلم : ميـــنا ألبير



مؤسسة اباتشى تحتفل بعيد ميلاد خدام الويب اباتشى الخامس عشر :

اعلنت مؤسسة اباتشى **Apache Software Foundation** عن احتفالها بخدام الويب المفتوح المصدر **Apache Web Server** وهو اكثر خوادم الويب استخداماً فى العالم .

بدأ المشروع من يوم ٢٣ من شهر فبراير عام ١٩٩٤ وكان عبارة عن

انقسام **fork** من مشروع **NSCA httpd** ليصبح المشروع الجديد تحت اسم اباتشى .
تم أصدار الأصدار الأول **Apache Server 1.0** فى شهر ديسمبر من عام ١٩٩٥ وتبعة الأصدار ١,٣ بعد أربع سنوات !!
تم الأعلان عن مؤسسة اباتشى **Apache Software Foundation** فى شهر مارس عام ١٩٩٩ للحصول على الشكل القانونى للمشروع بالأضافة الى الحصول على الدعم المالى اللازم .
وفقاً لأحصائيات مؤسسة اباتشى فان ١١٢ مليون موقع ويب يعتمد على خادم اباتشى وهى ما تمثل ٧٠% من مواقع الويب اجمع .
اخر الأصدار التحديثية للأصدار الأول من اباتشى **v1.x** صدر فى بداية الشهر الجارى .
اخر اصدار مستقرة منه صدرت فى ٥ أكتوبر ٢٠٠٩ وتحمل رقم الأصدار ٢,٢,١٤ .
خادم اباتشى يخضع تحت ترخيص الأصدار الثانى من رخصة اباتشى **Apache License v2**

<-->

مؤسسة البرمجيات الحرة Free Software Foundation تدعو جوجل لتحرير الويب من تقنيتى Flash و H.264

بعد اندماج شركة (**On2 Technologies** تقدم حلو لترميزات الفيديو) مع شركة جوجل بشكل نهائى ارسلت مؤسسة البرمجيات الحرة خطاب لجوجل تدعوها لجعل ترميز الفيديو **VP8** حر ليمكن الجميع استخدامه فى تطبيقات الويب بدون قيود وان تعتمدة فى موقع **YouTube** مما يبشر بتحرير الويب الترميزات الأحتكارية (والتى تتطلب رسوم) مثل **H.264** وايضا من البرمجيات الأحتكارية مثل **Flash** حتى يتسنى للجميع تصفح الويب بجميع محتواياته بشكل حر بالكامل .
وقد اوضحت مؤسسة البرمجيات الحرة **FSF** ان اذا قامت جوجل بهذه الخطوة فقد يساعد ذلك فى تصفح الويب بشكل كامل لمزيد من الأجهزة واستخدام التقنية لشريحة اكبر من البرمجيات .
اذا تم فعلا تحرير ترميز **VP8** ودمجة مع لغة **HTML 5** واستخدامة فى متصفح الويب جوجل كروم وتحويل الفيديوهات الموجودة على موقع **YouTube** ذلك الموت السريع للفيديو المعتمد على تقنية فلاش الاحتكارية .
كما اوضحت مؤسسة **FSF** ان اذا لم تستجب جوجل لهذا فانها بذلك ستؤكد انها لا يهمها تحرير الويب ولا تحرير مستخدمية ولكنه بذلك تكون تعمل على الهيمنة عليه ...



صدور واجهه Gnome Shell 2.29.0 مع نظام تنبيهات جديد :

الأن متوفر واجهه جنوم شيل Gnome Shell 2.29.0من اجل التجربة والأبلاغ بالعلل .

من الجدير ذكره ان هذا الأصدار يحتوى على نظام تنبيهات جديد new notification systemمدمج مع الواجهه وهو عبارة عن شريط سفلى ينبهك عند توصيل اجهزة ملحقة كما انه يوضح الصوتيات التى تعمل الأن فى الخلفية كما انه يمكنك الأطلاع على التنبيهات السابقة بضغطة زر .

كما انه تم اضافة امكانية تعديل حالتك online statusعلى برامج المحادثة كلها من ايقونة واحدة .

كما ان التأثيرات المستخدمة فى حالة التنقل بين النوافذ المفتوحة اصبحت اجمل .

مازال المشروع فى مرحلة مبكرة لذا لاينصح استخدامة على اجهزة العمل ...

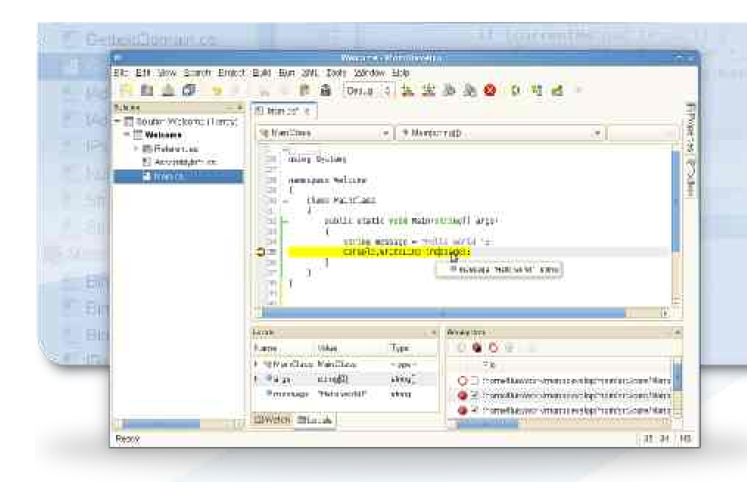
يمكنك تنزيل الكود المصدر للواجهه من خادم FTPالخاص بجنوم كما يمكن تجربتها على توزيعة اوبنتو من خلال تثبيتها من المستودعات التجريبية PPA

<-->

منصة التطوير MonoDevelop تدعم نظام MeeGoo الجديد :

اعلن Miguel de Icaza رئيس مشروع التطوير والبرمجة مونو Mono والذى ترعاه شركة Novell عن اضافة دعم لبرمجة التطبيقات بلغات ال . Net لمنصة MeGoo والتى تمثل اندماج لمشروعى Maemo من شركة نوكيا ومشروع MobiInمن شركة انتل والذى تم الأعلان عنه فى مؤتمر Mobile World Congressالمنعقد فى مدينة بارشلونة الأسبوع الماضى .
مشروع MeGooوهو عبارة عن نظام مفتوح المصدر بالكامل ومبنى على نواة لينوكس وموجهه للهواتف الجوالة الذكية ذات الأمكانيات العالية وايضاً للأجهزة اللوحية والنت بوك والعديد من الأجهزة الأخرى ...
هذه المنصة تستخدم بيئة QTلبرمجة الواجهه الرسومية لتطبيقاتها المختلفة كما ان سيتم توفير البرمجيات المختلفة لها عن طريق سوق Nokia's Ovi

وفقاً لتصريحات شركة نوكيا فان ٢٠% من هواتفها التى ستصدر عام ٢٠١١ ستعمل بهذا النظام .
اعلان De Icazaكان متوقعا حيث ان مشروع مونو اظهر اهتمام كبير بانظمة الهواتف من العام الماضى بعد اطلاقهم لمنصة MonoTouchلبرمجة التطبيقات لنظام ايفون ...
كما اعلن القائمين على مشروع مونوا نهم يعملون الأن على اطلاق منصة ستدعى MonoDroidوالتى ستتيح برمجة تطبيقات لنظام جوجل اندرويد .





صدور نواة جديدة من لينوكس Linux 2.6.33 الأصدار النهائى :

اعلن لانوس عن الأصدارة الجديدة من النواة والتى تحمل رقم الأصدر ٢,٦,٣٣ والتى تحمل لنا الآلاف من المزايا والتطويرات ومن اهمها اضافة تعريف مفتوح المصدر لكروت الشاشة Nvidia والذى يدعم خاصية KMS ولكن للأسف لا يدعم التسريع ثلاثى الأبعاد حتى الأن .. ولكن التعريف يحمل الكثير من المزايا والخصائص والتى تميزة الأن عن التعريف الحر المستخدم فى معظم التوزيعات حالياً .
كما تم أضافة دعم لتقنية ATA Trim والذى يعنى اداء افضل للأجهزة التى تعمل بالأقراص الثابته الصلبة . SSDs
كما تم أضافة لخاصية DRBD (Distributed Replicated Block Device) .
كما تم اضافة المزيد من التطويرات الى حلول الأجهزة التخيلية القائمة على النواة . KVM
وكما هو معتاد مع كل نواة جديدة يتم اضافة المزيد من التعريفات لدعم المزيد من العتاد حيث تم اضافة تعريفات جديدة مستخدمة فى برنامج الأجهزة التخيلية VMware HyberVisors وتعريفات لوحدات التخزين المختلفة والمزيد من كروت الشبكات اللاسلكية ...
اطار عمل ramzswap والمعروف سابقاً باسم compcache تم اضافة لمنطقة التدريج staging area . هذا الأطار يعمل على ضغط البيانات الموجودة على ذاكرة السواب (التبادل) الوهمية Virtual Swap وتخزينها على الذاكرة الفعلية العاملة working memory والذى يضيف مزيد من الفاعلية والسرعة .

اذا كنت احد مستخدمى لينوكس اللذين يعملون بالنواة الأفتراضية للتوزيعة بالتأكدي ستتمتع بالمزايا الجديدة للنواة الجديدة مع الأصدارات القادمة من توزيعتك المحبوبة وربما تحصل على المزيد من المزايا الممكن اضافاتها فى الأصدار التحديثية للنواة والتى ستصدر الأسابيع القادمة .

صدور Blender 2.5 الأصدارة التجريبة الفا ١ :



اعلن مطورى برنامج بليندر -اقوى منصة تصميم ثلاثى الأبعاد مفتوحة المصدر- عن الأصدارة التجريبية الفا١ والتى تمثل ثانى خطوة فى الطريق للأصدارة النهائية والتى ستحمل الأصدار ٢,٦ والمنتظر صدورها فى منتصف العام الجارى ..
هذا الأصدار يحمل العديد من المزايا الجديدة بالأضافة الى اصلاح اكثر من ١٠٠ علة برمجية مقارنة بالأصدار السابق الفا٠ .
تشمل هذه الأصدارة واجهه رسومية جديدة GUI تحمل ايقونات جديدة واختصارات مخصصة جديدة للوحة المفاتيح Keyboard Shortcuts وتغييرات وتعديلات فى المعمارية الداخلية للبرنامج ..
منذ زمن وبرنامج بليندر متوفر باصدارة للمعالجات ذات المعمارية Bit ٦٤ لكل من لينكس ووينذوز ولكن هذا الأصدار يضيف دعم أيضا لنظام ماك .

لا ينصح باستخدام هذا الأصدار على اجهزة العمل حيث انه مازال غير مستقر .

البرنامج تحت رخصة جنو العمومية الأصدار الثانى GNU GPL v2 كما انه مدعوم من مؤسسة بليندر Blender Foundation غير الربحية .

لأنك فيدورا (شعر)

بقلم : مصعب الزعبي

زرقاء العين في حاسبي
وعليه لن أجلس بعد مقهورا [١]
لن أتعب بعدك يا فيدورا [١]
حبيبتي أنت
معشوقتي أنت
دمت لي يا أمورة [١]
***
دعوت عاشقات مدينتي
أبلغهن حبي الجديد
قلبي الوليد
أعرفهن على السنيورة
***
كنت وحيدا
كنت حزينا
على فمي رسمت بسمتي
فغدوت مسرورا
***
آنستني في وحشتي
في غربتي
في ليلتي
في حاسبي ، في موقعي أمنتني
بعد أن كنت قبلك مذعورا
***
كنت مواليا لغيرك
لعدوك الغيتسي
منافسك البرتقالي
أمك قبعة البندورة
***
أصارحك أني أحبك
وأحب مستخدمك
وأقنع غيره بك
لأنك الأفضل
لأنك الشطورة
لأنك حبيبتي
لأنك فيدورا ،،،

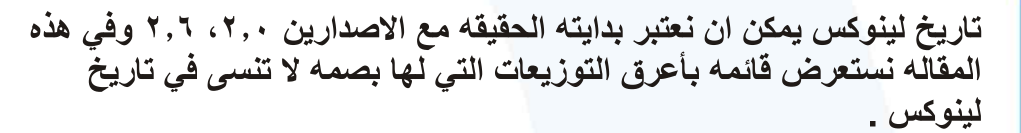
---

(١) تخضع لمشيئة الله وحده ،

خمس توزيعات غيرت وجه لينوكس

The Five Distros That Changed Linux

بقلم : فتحي محمد القدسي



تاريخ لينوكس يمكن ان نعتبر بدايته الحقيقه مع الاصدارين ٢,٠، ٢,٦ وفي هذه المقاله نستعرض قائمه بأعرق التوزيعات التي لها بصمه لا تنسى في تاريخ لينوكس .

بالإمكان قياس تاريخ لينوكس بعده طرق في الغالب عن طريق إرقام الإصدارات .

بدايه نواه لينوكس في سبتمبر ١٩٩١ مع الاصداره رقم ٠,٠١ وهناك طريقه اخرى بالطبع للنظر في عمليه تطور لينوكس وهي الشروط المطبقه في توزيعاته الهامه .

بالنسبه للمستخدمين هذه التوزيعات تعتبر قمه الجبل وماخفي كان أعظم فبإمكانك ان تقارن او تناقش أفضليه توزيعه على أخرى لكن كل ذلك ليس مكانه هنا والسبب ان قائمتي التي اخترتها غيرت نظرتنا وطريقه استخدامنا للينوكس .

بنيت هذه القائمه بناء على النظر في تاريخ لينوكس بالاضافه الى تجربتي الشخصيه بالرغم من اني لم اكن مستخدما للينوكس في سنواته المبكره والسبب اني كنت في ذالك الوقت اعمل مع مجتمع يونكس في مشروع نظام التشغيل BSD والذي سرعان ماغادرته .

ولنبدأ في قائمتنا ويتربع على راس القائمه :

<-->

سلاكوير

Slackware



1993 البدايه الحقيقيه لأول توزيعه شعبيه

وهذه التوزيعه شدتني انا وكثيرين الى عالم لينوكس في بدايه التسعينات مؤسسها **Patrick Volkerding** واخذ اسمها من **Church of the SubGenius** الموقع الاشهر في عالم الهاكر في التسعينات في البدايه كانت سلاك عباره عن مشروع جانبي ولكن بسرعه تحولت الى اكثر من مجرد توزيعه صغيره مع اسم طريف احب العديدين تجربه لينوكس لكن لم تكن لديهم الخبره الكافيه في دائره لينوكس المرعبه لكثيرين **build/make/compile** فكل شئ عباره عن ملفات مصدريه ويجب عليك بناءها بنفسك وسلاك اليوم مازالت بالرغم من هذا التطور والطفره التي حصلت في عالم لينوكس غير مناسبه للمستخدم المبتدئ لكنها كانت البدايه لمن اراد ان يعرف لينوكس.

بدايه سلاك كتوزيعه غريبه نوعا ما فـ **Volkerding** لم يكن يخطط ابدا لعمل توزيعه ،مالذي حصل اذا ؟ كما قال في مقابله له في ١٩٩٤ ان استاذه طلب منه ان يشرح له كيفيه تثبيت لينوكس على جهازه الشخصي في المنزل ومن ثم شاركه هذا الامر بعض تلامذته الذين كانو عليهم بعض الاعمال في **LISP** لذلك يقول **Volkerding** ذهبنا الى معمل الحاسب وقمت بتثبيت **SLS (Soft Landing Systems)** نسخه من لينوكس .

هناك العديد من التوزيعات مثل **SLS (Soft Landing Systems)** الانفه الذكر ومن قبلها **Yggdrasil** التي نستطيع ان نقول انها اول توزيعه لينوكس تجاريه سبقت **slackware** بالرغم من ذلك كله ببساطه عملت بشكل افضل من بقيه التوزيعات الاسبق منها وحجزت لنفسها مكانا في قلوب مستخدمي لينوكس الاقدمين .

ديبيان
Debian



مرحبا بمجتمع لينوكس

بينما Slackware جذبت المستخدمين الى لينوكس بالآلاف فان Ian Murdock والذي تخرج من جامعه Purdue بدء العمل في مجتمع لينوكسي جديد ظهر الى الوجود باسم Debian .

بعض التوزيعات القديمه بمافيها slackware كانت ثمره جهود مجموعه من المطورين الملهمين بينما نجد توزيعات اخرى مثل Caldera, Red Hat,Yggdrasil . والتي هي توزيعات تجاريه في الاساس كانت تقوم بجهود موظفين في هذه الشركات Murdock. كان له وجه نظر اخرى هي عباره عن فكره اخرى وضحها في The Debian Manifesto حين قال "دبيان هي فرع جديد من توزيعات لينوكس لا تقتصر على مطور واحد منعزل او مجموعه واحده كما هو حال بقيه التوزيعات في الماضي بل هي اكثر من ذلك، دبيان تطور بشكل مفتوح على الجميع تحت فلسفه Linux وGNU "

وكان على حق دبيان كانت بالفعل فرع جديد من توزيعات لينوكس واليوم مع openSUSE, Fedora, Ubuntu تغير مفهوم توزيعات لينوكس التجاريه واصبح لها مفهوم جديد كما انها لها جذور عميقه في مجتمع لينوكس بالفعل كانت فكره جوهريه واصبحت أساسا بنيت عليه الكثير من الاشياء فيما بعد.

نعم كانت linux kernel تطور من قبل مجتمع مفتوح عبر القوائم البريديه ومجموعات اليوزنت والمستودعات لكن هناك الكثير من الاشياء الصغيره التي تخص البرامج كانت بحاجه للمجتمع والذي كان فكره جديده وبالفعل ففكره العمل التطوعي في عالم لينوكس لاقت الكثير من الاستحسان وانخرط فيها الكثير .

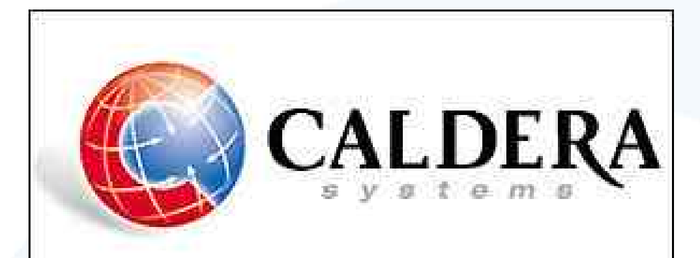
ولكن تأتي الرياح بما لا تشتهي السفن، فسرعان ماقامت حرب ضروس بين المجتمع ومؤسسه
http://practical-tech.com/operating-system/when-is-debian-not-debian/

كذلك الحال في مجموعات المصادر المفتوحه مثل Mozilla ضد فايرفوكس
http://practical-tech.com/operating-system/linux/fox-wars-debian-vs-mozilla

وقامت خلافات اخرى في داخل التوزيعات نفسها حول معايير عمل توزيعه جديده
http://practical-tech.com/operating-system/disgruntled-delay-etch

لكن بالرغم من كل هذه المعارك الجانبيه استطاعت Debian بطريقه او بأخرى ان تكون لها قصب السبق وتسيطر على قمه هرم توزيعات لينوكس .
كل مجتمعات لينوكس تدين بالفضل ل Debian في دورها الطلائعي . والى اليوم مازالت Debian من اكثر التوزيعات شعبيه في العالم . العديد من التوزيعات بما فيها Ubuntu, MEPIS, Knoppix, Xandros تعتمد على الكود المصدري لدبيان .

من نافله القول اذاً ان نخرج بنتيجه وهي انه بدون Debian فانه لم يكن ليقوم لمجتمع لينوكس قائمه



كالديرا
caldera 1993/4

اول توزيعه خاصه بالأعمال

الان عندما تذكر caldera يذكر تحولها الى SCO الد اعداء
لينوكس جنبا الى جنب مع ميكروسوفت .لكن بدايتها لم تكن كذلك ابدا.
اذا ماهي القصه الحقيقيه وراء هذه التوزيعه العريقه ؟

في البدايه رأى مؤسسيها Ransom و Brian Sparks ان لينوكس ممكن ان يكون شيئا عظيما كنظام تشغيل خاص بقطاع الاعمال ومن ناحيه اخرى سيكون طريقه فريده لمحاربه ويندوز في عالم الكمبيوتر المكتبي والسرفرات وكما صرح Love في كتابه The Love of Linux بدايه Caldera كانت كمشروع مشترك انبثق من Novell والهدف منه محاربه ميكروسوفت ونظامها العتيد في ذالك الوقت Windows NT .
في هذه الفتره حاولت ميكروسوفت ان تستحوذ على اكبر حصه من سوق انظمه الحاسب وتقصى نوفل بعيدا عن السوق يقول Love بانه احس هو و Bryan Rob, بان بامكانهم الكثير في مجال انظمه التشغيل ولن يكلف ذلك ملايين الدولارات اذا تم التركيزعلى مجموعه صغيره من المهندسين تهتم بالواجهه الرسوميه وخدمات السرفر .

ونظرا لجهود المجتمع القائم على التطوع واعتماد فكره مجتمع لينوكس المفتوح القائم على فلسفه المصادر المفتوحه كل ذلك أدى الى انخفاض التكاليف اللازمه لتطوير انظمه التشغيل وظهور فكره سطح المكتب هناك ايضا ال backend services التي دعمتها NetWare وبذلك استطعنا -يقول love -إجبار ميكروسوفت على اعتماد NetWare services في انظمتها المكتبيه ولتفتح الطريق امام المنافسه .

لم يتغير شئ من فتره طويله بل الاكثر من ذلك هو عندما اقدمت ميكروسوفت على الاعلان عن تخليها عن دعم اجهزه netbooks في نظام xp في المقابل نرى ان خطه IBM في مجال الاعمال باستخدام لينوكس
http://practical-tech.com/operating-system/ibm-and-linux-the-early-years

وسنرى تعليق Love على فكره ان تكاليف بناء نظام تشغيل انخفضت بشكل كبير باعتمادها على فلسفه المصادر المفتوحه.
لسوء حظ Novell فبينما كان المؤسس Ray Noorda يدعم الفكره ،توقفت Novell عن دعم لينوكس ، كانت بالفعل امر فظيع خاصه انها استمرت طوال العقد الاخير في مجال لينوكس وتعرف دقائقه ولعبت دور كبيره في قوه لينوكس الحاليه لكنها اختارت طريقا اخر buying SUSE .

لذلك قام كلاً من Love و sparks بتأسيس Caldera وبالاتفاق مع Love استكملا ما بدآه بإنشاء نظام تشغيل بديل في المجال الصناعي والتجاري هذا النظام سيسمى فيما بعد Caldera Network Desktop وذلك في نهايات ١٩٩٥ وكان مبنى بالأساس على Visix مع تعديلات وتطويرات عليه .

مالايعرفه كثير من الناس هو ان مطوري Caldera قاموا بذلك بمساعده من Red Hat لكن بالرغم من ذلك Caldera الان على مفترق طرق فهي تحتاج ان تسيطر على كل صغير ه وكبيره لتؤكد لقطاع الاعمال جوده وتفرد منتجاتها .
Red Hat كانت تعرف في ذلك الوقت ان من يشتري نظام لينوكس هم الهاكرز والمطورون وذلك بسبب التحديثات السريعه للكود المصدري .
Caldera مازالت حتى هذه اللحظه تقف وراء حلمها بالوصول الى ايصال فكره ان لينوكس سينتشر تجاريا وسيكون مربح اقتصاديا ايضا .
في نهاية التسعينات اتفق كلا من Caldera و Red Hat على صفقه business buyers التي قضت على احلام Caldera استمر Noorda في الاستثمار في caldera لكن تدهورت صحته فقام المدراء الماليين للمجموعه canopy بالإقدام على قرار كان من شأنه ان يقضى على البقيه الباقيه من احلام caldera وكان هو ابرام صفقه

the purchase of SCO، كان قرارهم هو الطريقه الأسرع للثراء طبعا النتيجه في آخر الأمر هي الفشل الفظيع لـ SCO/Caledera وهكذا نرى ان caldera وقبل قرارها الانتحاري كانت تمثل خطوه وحجر اساس للمشروع التجاري القائم على لينوكس والذي كان سيكون له حصه لايستهان بها اليوم في سوق الاعمال بالطبع Red Hat تعلمت من الدرس جيدا والنتيجه انها استطاعت ان تنقل نفسها من توزيعه تقتصر على الهاكر الى توزيعه يستفيد منها قطاع الاعمال ايضا .

<-->

ريد هات

Red Hat Enterprise Linux 2.1AS

عندما يدخل لينوكس عالم الشركات

في عام ٢٠٠٤ كره مستخدمي لينوكس Red Hat. لماذا ؟
لانها اعلنت انها ستجعل نسخه Red Hat 9 حصريا على القطاع التجاري وستحرم المستخدمين العاديين منها .
لماذا خرجت Red Hat من كونها تخدم المستخدمين العاديين والهاكر ايضا الى خدمه رجال الاعمال فقط ؟
لأنهم ادركو ان النقود لم تعد تكفي للاستمرار على نفس النهج في السنوات القادمه .

في مقابله له على الايميل صرح Michael Tiemann رجل ريدهات المختص بشؤون المصادر المفتوحه بأن هناك فرص هائله لمساعده الزبائن في تقليل الكلفه من ٥٠%الى ٩٥% مع زياده الجوده من ٥٠% الى ١٠٠% ونحن مستعدون تماما لذلك مع قدرتنا ايضا على المشاركه في تطوير المصادر المفتوحه وبطريقه فعاله ..

بالإضافه الى ذلك فإن الزبائن يستطيعون ان ينتقلوا من نطاق الانظمه المملوكه وبرامجها الى عالم المصادر المفتوحه بدون ان ينفقوا مبالغ طائله وسنوفر عليهم الوقت الذي سيقضونه في صيانه انظمتهم المفتوحه وكنتيجه لما سبق يمكن القول ان كلا الفريقين في هذه المعادله سواء الزبائن او الشركاء في العمليه التطويريه وهذا جعل ريدهات انتربرايز تتبوأ مكانا مرموقا في مجال الاعمال التجاريه القائمه على لينوكس.

واضاف اذا كان بإمكانك عمل ذلك فبإمكان ريدهات عمله ايضا وهذا ليس تفاخرا بل هو واقع الحال فريدهات في السنه الماليه الماضيه حققت اكثر من نصف بليون دولار في الوقت الذي يعاني في الجميع نجد ريدهات تنهض وتواصل نموها المتطرد.

فريدهات استطاعت ان تنقل نفسها من لينوكس يقتصر على الهاكر الى لينوكس يخدم قطاع الاعمال وبفعاليه تامه وهي بذلك وضعت بصمتها في التاريخ كما انها مع ذلك كله مازالت تدعم مجتمع لينوكس كما تدعمه دبيان وذلك عبر توزيعتها فيدورا والتي تقف وراءها بكل قوه وستستمر في ذلك.

وفي نهايه كلامه وضع مقارنه عن الفرق الجوهري بين فيدورا وريدهات انتربرايز حين قال "فيدورا هي :كيف تكون هناك وانتربرايز هي كيف تبقى هناك"



أوبنتو
ubuntu 2004

لينوكس للجميع

رائينا ان لينوكس نمت وزادت شعبيتها بين المبرمجين وقطاع الاعمال لذلك ظهرت حاجه ماسه لتسهيل استخدام لينوكس للشخص العادي والبسيط وهنا انبرى لهذه المهمه الجليله Mark Shuttelworth وقرر ان يعمل شئيا للمساهمه في هذه المسئله وكما لا يخفي على اغلبنا ان مارك شتالورث ملتي مليونير اي ان لديه المصادر الكافيه لعمل هذا الشئ .وبمانه لديه علاقات مع دبيان قرر ان يعمل مع ما سيطلق عليه فيما بعد اسم مجتمع ابونتو وكان قراره عمل توزيعه مبنيه على دبيان لكنها موجهه للمستخدم العادي لذلك فيها تسهيل في عمليه تثبيت النظام وكذالك طريقه التعامل مع البرامج واحتوائها على حزم جاهزه مثتيته مسبقا سيكون اصدار هذه التوزيعه كل سته اشهر على ان تتمتع بكافه التحديثات الامنيه خلال ثمانيه عشر شهرا ودعم تقني يرافق كل اصداره .

احلام كبيره لكن خلفيه شتالورث الماليه ساهمت في تحقيق هذه الاحلام واصبح مجتمع ابونتو حقيقه واقعه .

تدرجت ابونتو في شهرتها حتى وصلت اليوم وباتفاق الجميع الى اشهر توزيعات لينوكس على مستوى العالم ومنذ ٢٠٠٥ اصبحت تحتل المراكز الاول في موقع مراقبه التوزيعات الاشهر . distrowatch

حظيت ابونتو بكثير من الدعم من قبل المصنعين وعلى رأسهم شركه ديل اما ٢٠٠٩ فقد شهد نقله نوعيه بالنسبه لابنتو فقد اصبحت شعبيتها تضرب الافاق لدرجه ان الكثير من المستخدمين الجدد لم يعد يميز مابين لينوكس وابونتو فلينوكس هو ابونتو وابونتو هو لينوكس .

....

اخيرا نخرج بالخلاصه التاليه

سلاك وير:اول من نشر لينوكس وعرف العالم به

دبيان:اول مجتمع لينوكس

كالديرا:اول توزيعه اظهرت امكانيه لينوكس في دخول عالم الاعمال

ريدهات انتربرايز:اول توزيعه وضعت لينوكس في سوق الاعمال فعليا

ابونتو:اول توزيعه سهلت استخدام لينوكس للجميع

....

هذا عن قائمتي ماذا عن قائمتك انت؟

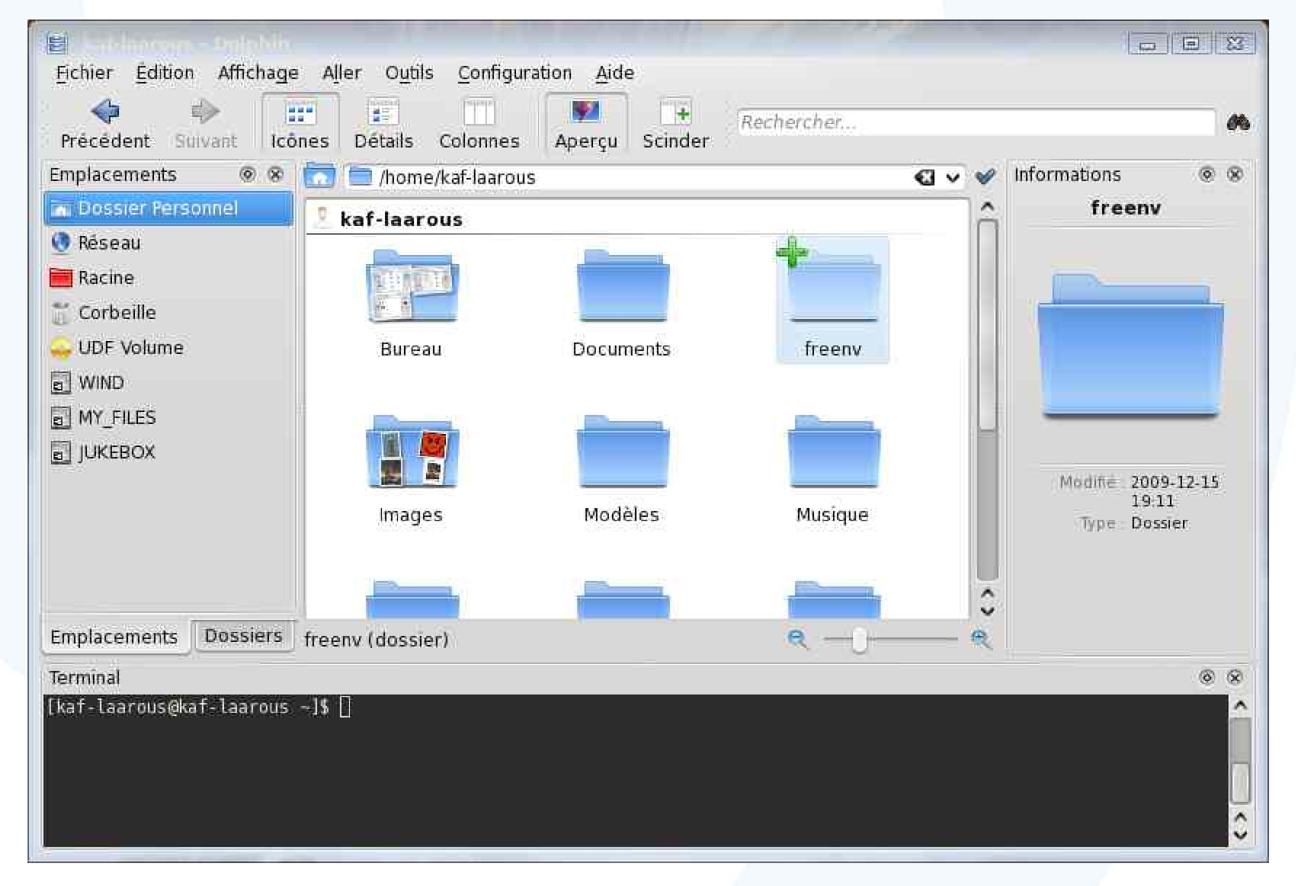
مدير الملفات Dolphin

بقلم : ياســر يكن

Dolphinهو مدير الملفات الذي يأتي افتراضيا مع سطح المكتب KDE4، وقد تم استبداله مكان konqueror الذي أصبح متصفحا للمواقع على KDE4، طبعا ويمكن استعماله كمتفح للملفات أيضا ...

الواجهة

هذه هي واجهة مدير الملفات Dolphin



نلاحظ في مدير الملفات ما يلي :

* القوائم في الأعلى
* شريط الأيقونات في الأسفل مباشرة
* الشريط الخاص بالمجلدات في الجهة اليسرى
* أسفله مباشرة نجد الشريط الخاص بالمكان المتواجد فيه
* النافذة الخاصة بالمعلومات عن المجلد الذي يتم تحديده

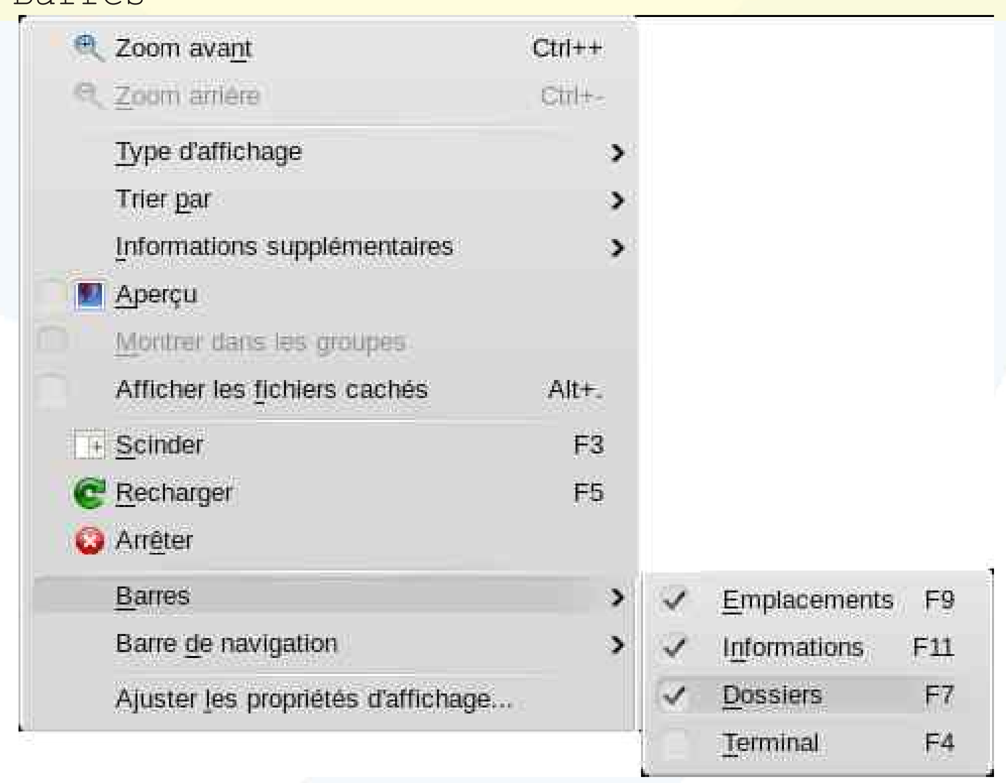
وهذا ليس الشكل الوحيد للمدير، لأنه بالإمكان تغير أماكن الأشرطة عن طريق سحبها بواسطة الفأرة وإفلاتها في المكان المرغوب فيه ...

توجد أشرطة وقوائم أخرى بالإمكان التحكم في إظهار وإخفاء ما نريد منها في القائمة :

```
Affichage >Panneau
```

أو

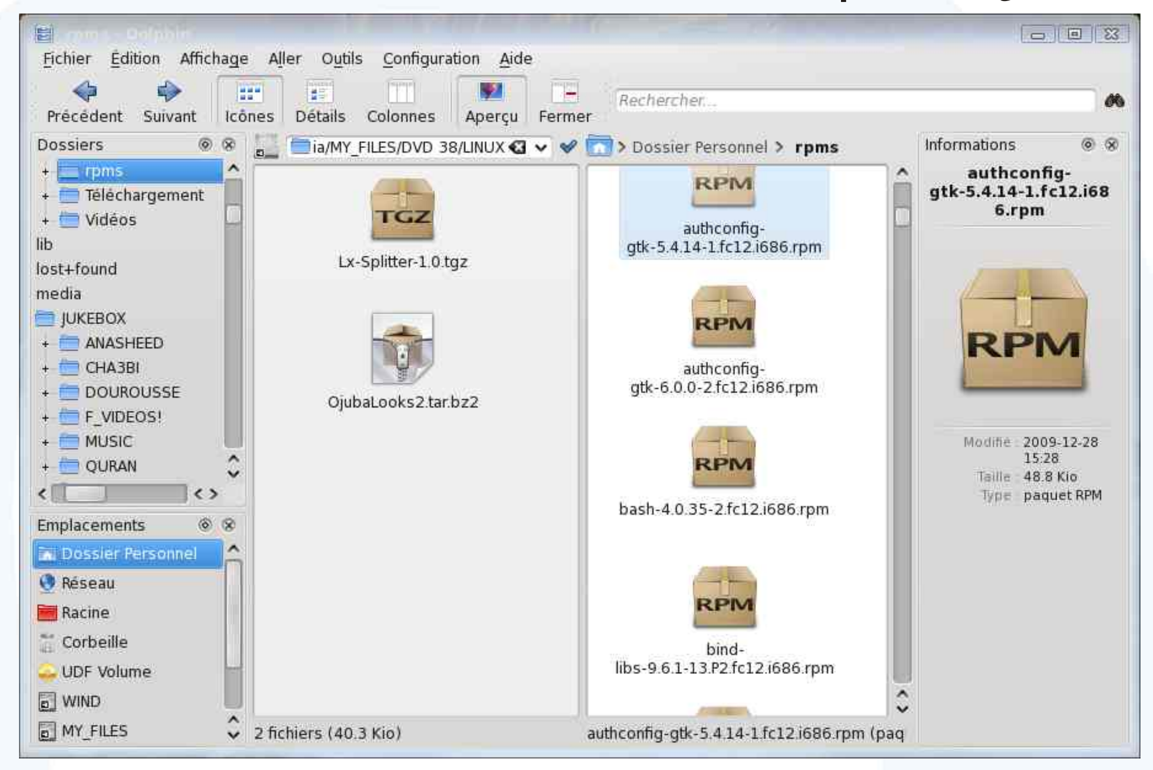
```
Afficher >Barres
```



تنبيه !

إذا كان شريط القوائم غير ظاهر أو تم حذفه خطأ، فبالإمكان استعادته عن طريق النقر بزر الفأرة الأيمن في مكان فارغ من مدير الملفات Dolphinثم اختيار استعادة شريط القوائم، أو الضغط على مفتاحي Ctrl+mمن لوحة المفاتيح .

بالإمكان مع Dolphin أن تقسم نافذة العمل إلى قسمين لتسهيل العمل كالنسخ والنقل من النافذة الأولى إلى الثانية والعكس وذلك بالنقر على أيقونة Split أو Scinder



وبالنقر على أيقونة Colonne تنقسم النافذة إلى أعمدة تفاعلية يعرض كل واحد منها محتويات المجلد المحدد في العمود على اليسار عند النقر عليها .

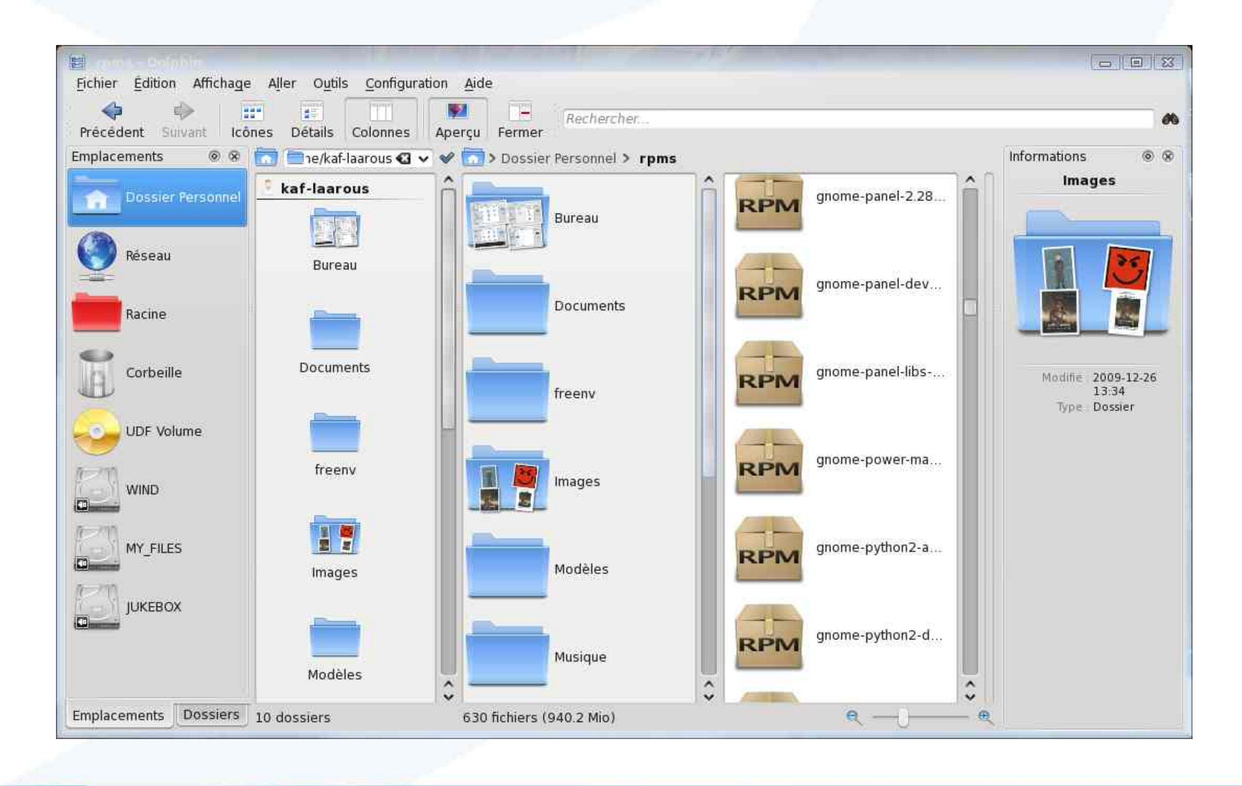
وبالإمكان إضافة الأيقونات المفضلة في القائمة الخاصة بالأيقونات من لسان القائمة :

```
Configuration > Configurer les barres d'outils
```

ثم تحديد الأيقونات المراد ظهورها في القائمة .

استعمال Dolphin

نسخ ونقل الملفات :



الإستخدام المفضل هو وضع Split أو Scinder، وبالنقر على الملف المراد إجرا العملية عليه وسحبه وإفلاته في المكان المرغوب فيه ثم الإختيار من القائمة التي ستظهر : نسخ أو نقل أو إنشاء رابط " وصلة "

عكس الإختيار :

من القائمة

```
Edition > Inverser la sélection
```

الترتيب :

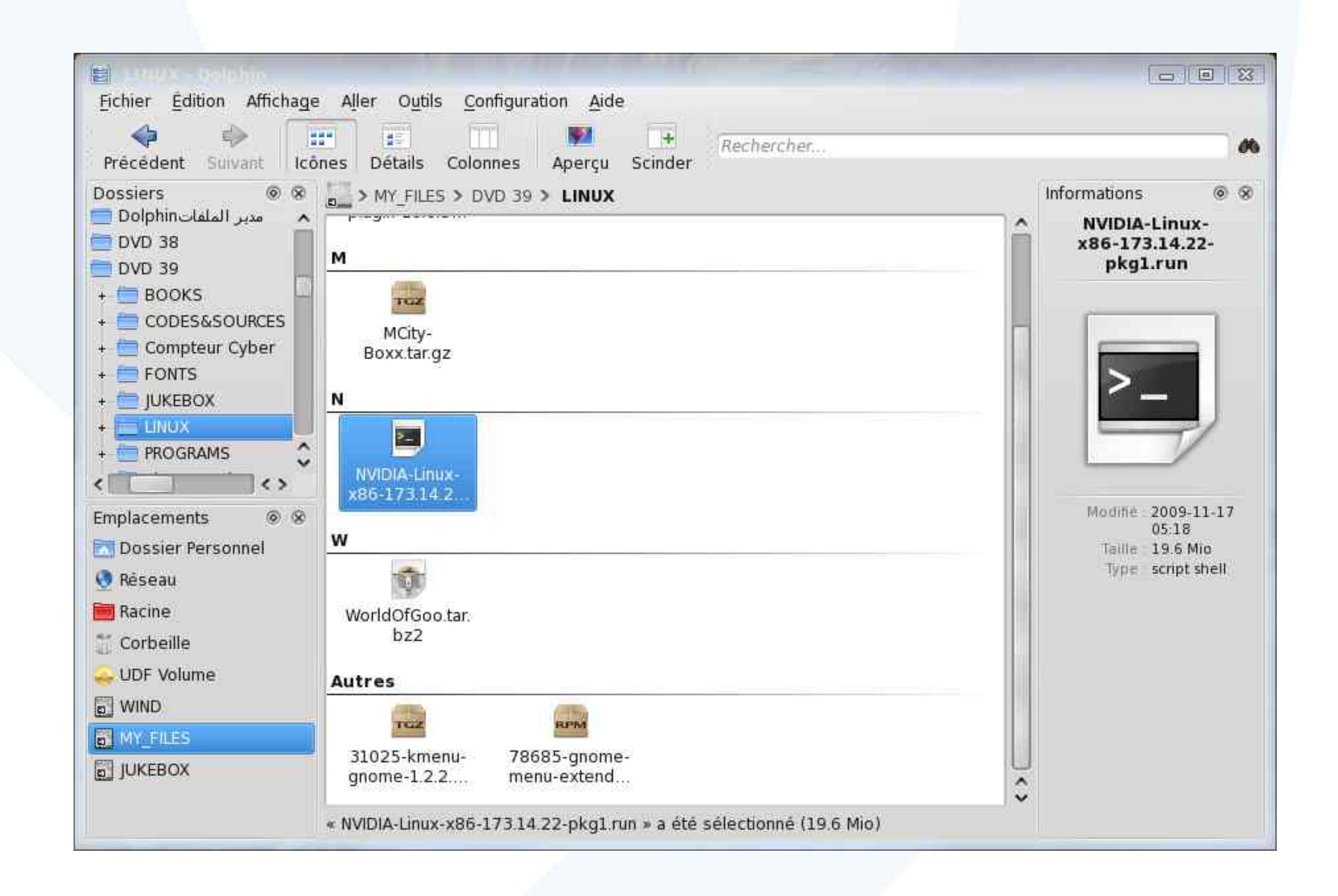
من القائمة

```
Afficher > Trier par
```

المجموعات :

بالإمكان اختيار ترتيب الملفات والمجلدات على شكل مجموعات وذلك من القائمة

```
Afficher >Montrer dans les groupes
```



مدير الحزم
Zypper

بقلم : مصطفى رفاعي

Zypp (libzypp) : هو محرك إدارة حزم يشغل العديد من التطبيقات الأخرى مثل Yast , Zypper و PackageKit الخاص بـ OpenSUSE/SLE .

Zypper : هو واجهة سطر الأوامر لـ libzypp ليجعل مميزاته – مثل دخول المستودع و قرائتها ، حل الإعتماديات ، تثبيت الحزم ... إلخ – متاحة عبر سطر الأوامر .

مميزات Zypper

التالي بشكل رئيسي هو قائمة مميزات zypper كواجهة سطر أوامر بالإضافة لقائمة مميزات libzypp التي أتيحت بواسطة Zypper . قائمة المميزات هذه تتوافق مع أحدث إصدار Zypper للمستخدمين .

** إدارة الحزم

* تثبيت/إزالة الحزم بالإسم أو بالوظيفة التي تقدمها .
* تثبيت/إزالة إصدارة معينة من الحزمة .
* تثبيت ملفات RPM و إذا كان ينقصها إعتماديات فيقوم بجلبها من المستودعات .
* تثبيت و إزالة حزم متعددة بسطر واحد [بإستخدام +/- او ~/!] .
* تحديد مستودع معين لكل حزمة بإسباق إسم الحزمة بإسم المستودع .
* تحديث كل الحزم المثبتة بإصدارات أحدث عند توافرها .
* تثبيت حزم/قوالب/رقع/منتجات .
* تقديم ملخص تثبيت شامل .
* إمكانية تجنب تثبيت الحزم الموصى بها – فقط المطلوبة - .

** إدارة المستودعات

* إضافة/إزالة/إستيراد/تصدير المستودعات بكل سهولة .
* إستخدام مستودع ما بشكل مؤقت .
* قصر العمل على مستودعات معينة .
* تعديل مستودعات متعددة بمرة واحدة – إختيار بنوع الوسط أو المحلية أو الكل .

** الإستعلامات

* البحث عن حزم بالإسم أو الملخص أو المفصل .
* البحث عن حزم تطابق جملة فرعية ، تعبيرات شمولية شاملة القيمة (wild-cards) جميع الكلمات ، نفس الجملة .
* تصنيف نتائج البحث بإسم الحزمة و نوعها أو عرض كل إصدارات الحزم المتاحة من جميع المستودعات .
* عرض معلومات مفصلة عن حزمة/رقعة/قالب/منتج .
* عرض جميع الحزم/الرقع/القوالب/المنتجات المتاحة .

** إدارة التحديث

* عرض الرقع الضرورية أو كلها .
* تثبيت الرقع .

** أقفال الحزم

* إضافة أقفال للحزم بسهولة .. الإقفال بالإسم – يمكن استخدام التعبيرات الشمولية , – wild-cardsبنوع الحزمة ، بالمستودع .
* عرض الأقفال بالرقم أو بالإسم .
* إزالة الأقفال بالرقم أو بالإسم .

** الخدمات

* التحقق من الإعتماديات الخاصة بالحزم المثبتة .
* تثبيت الحزم الموصى بها من قبل حزم أخرى مثبتة .

** عام

* نمط غير تفاعلي لإستخدامه في السكربتات .
* ثلاثة مستويات من محتوى المخرجات .

إستخدام Zypper

**هذه قائمة بالأوامر الأكثر إستعمالا بالغالب :

عرض قائمة الأوامر و الخيارات المتاحة	Zypper
عرض المساعدة الخاصة بالأمر 'search'	Zypper help search
عرض قائمة رقع التحديثات الضرورية	Zypper lp
تثبيت الرقع الضرورية	Zypper patch
البحث عن الحزمة 'sqlite'	Zypper se sqlite
إزالة الحزمة 'sqlite'	Zypper rm sqlite
تثبيت الحزمة 'sqlite'	Zypper in sqlite
تثبيت جميع الحزم التي تبدأ بـ'yast'	Zypper in yast*
تحديث جميع الحزم لآخر إصدار متاح	Zypper up

** التركيب العام لجملة أمر zypperهي كالتالي :
zypper [خيارات-شاملة] أمر [خيارات-أمر] [متغير]

المكونات بين الأقواس ليست أساسية لذا فإن أبسط طريقة لتنفيذ عملية بـ zypperهي كتابة zypperملحوقا بالأمر . كمثال : لتثبيت الرقع المطلوبة للنظام قم بكتابة التالي : zypper patch . بالإضافة لذلك يمكنك إختيار خيار شامل global-optionأو أكثر بكتابتهم قبل الأمر . مثال : " zypper –non-interactive patch .قم بتثبيت الرقع الضرورية و لا تسألني عن شئ .. قرر بنفسك "
و لإستخدام خيار أو أكثر خاص بأمر معين command-optionقم بكتابتهم بعد الأمر . مثال : zypper patch “ –auto-agree-with-licensesقم بتثبيت الرقع الضرورية و لا تسألني عن الرخص فأنا موافق عليها جميعا ”
. بعض الأوامر تطلب متغير أو أكثر . مثال : “ zypper install mplayer .تثبيت الحزمة 'mplayer' ”

** إدارة المستودعات

عرض المستودعات الحالية	zypper lr أوZypper repos
عرض المستودعات الحالية بالإضافة لعناوينها	Zypper lr -u
عرض المستودعات و أولوياتها و ترتيبها حسب ذلك	Zypper lr -P
تحديث المستودعات	zypper ref أوZypper refresh
تحديث المستودع 'upd'	Zypper ref upd
تحديث المستودع 'upd' بالقوة	Zypper ref -f upd
التعديل على المستودعات	zypper mr أوZypper modifyrepo
تعطيل المستودع ٦	Zypper mr -d 6
تفعيل التحديث التلقائي و التخزين المؤقت لملفات الـ RPMللمستودع ' upd' تعيين أولويته إلى ٧٠	Zypper mr -rk -p 70
تعطيل التخزين المؤقت لملفات الـ RPMالتي تم تنزيلها من جميع المستودعات	Zypper mr -Ka
تفعيل التخزين المؤقت لملفات الـ RPMالتي تم تنزيلها من جميع المستودعات الخارجية	Zypper mr -kt
إضافة المستودعات	zypper ar أوZypper addrepo
إضافة المستودع update	Zypper addrepo http://download.opensuse.org /update/11.1/update update
إزالة المستودعات	zypper rr أوZypper removerepo
إزالة المستودع 'upd'	Zypper rr upd
إعادة تسمية المستودعات	zypper nr أوZypper namerepo
إعادة تسمية المستودع "٣" إلى 'upd'	Zypper nr 3 upd
إستيراد/تصدير المستودعات	zypper lr -e أوZypper repos –export
التصدير	Zypper lr –export backups/repos/foo.repo
الإستيراد	Zypper ar backups/repos/foo.repo

** إدارة الحزم

يتم إختيار الحزم – للتثبيت أو الإزالة أو غيرها – بأكثر من طريقة ممكنة مثلا :

* بالإسم : zypper in 'perl(Log::Log4perl)' أو zypper in qt
* بالإسم و/او المعمارية و/أو الإصدار : zypper in 'zypper<0.12.10 ' أو zypper in zypper.i586=0.12.11
* بإسم الحزمة و تحديد المستودع : zypper in factory:ftp
* بإسم ملف RPM الحزمة : . zypper in skype-2.0.0.72-suse.i586.rpm

تثبيت الحزم	Zypper install أو zypper in
تثبيت الحزمة "git"	Zypper in git
تثبيت الحزمة "MozillaFirefox"	Zypper in MozillaFirefox\<3
تثبيت كل الحزم التي تبدأ بـ"yast"	Zypper in 'yast*'
تثبيت القالب "lamp_server"	Zypper in -t pattern lamp_server
تثبيت الحزمة vim و إزالة الحزمة emacs	Zypper in vim -emacs
تثبيت الحزمة ' libxine1' من المستودع ' upd' و تثبيت الحزمة "amarok"	Zypper in amarok upd:libxine1
إزالة الحزم	Zypper remove أو zypper rm
إزالة الحزمة "sqlite"	Zypper rm sqlite
بناء الإعتماديات و تثبيت الحزم المصدرية	Zypper source-install أو zypper si
تثبيت الحزمة المصدرية ' smart' بإعتمادياتها	Zypper si smart
تثبيت الحزمة المصدرية ' smart' وحدها فقط	Zypper in -D smart
تثبيت إعتماديات الحزمة المصدرية ' smart' وحدها فقط	Zypper in -d smart

** إدارة الحزم

عرض الرقع الضرورية	Zypper list-patches أو zypper lp
تثبيت الرقع	Zypper patch
عرض جميع الرقع	Zypper patches
التحقق من الرقع	Zypper patch-check أو zypper pchk
عرض معلومات عن الرقع	Zypper patch-info
عرض معلومات عن الرقع	Zypper info -t patch
عرض تحديثات الحزم المتاحة	Zypper list-updates أو zypper lu
تثبيت تحديثات الحزم المتاحة	Zypper update أو zypper up

** الإستعلامات

البحث عن الحزم	Zypper searchاو zypper se
البحث عن الإسم المختصر RSIو مراعاة حالة الحروف ، أيضا البحث بالملخص و الوصف	Zypper se -dC –match-words RSI
البحث عن كل الحزم التي تبدأ بالكلمة "yast	Zypper se 'yast*'
عرض كل الحزم الموجودة بالمستودع "upd	Zypper se -r upd
عرض كل حزم ' 'sqliteالمثبتة	Zypper se -I sqlite
عرض كل القوالب المتاحة بالمستودع "upd	Zypper se -t pattern -r upd
جلب معلومات عن الحزم	Zypper infoاو zypper if
عرض معلومات عن الحزمة "amarok	Zypper info amarok
عرض معلومات عن الرقعة "amarok	Zypper info -t patch amarok
عرض معلومات عن الرقعة "amarok	Zypper patch-info amarok
عرض معلومات عن القالب "lamp_server	Zypper info -t pattern lamp_server
جلب معلومات عن الإعتماديات	Zypper what-providesاو zypper wp
عرض معلومات عن إعتماديات الحزمة "firefox	Zypper wp firefox

** أقفال الحزم

قفل حزم	Zypper addlockاو zypper al
قفل كل الحزم التي تبدأ بـ"yast2	Zypper al 'yast2*'
إزالة الأقفال	Zypper removelockاو zypper rl
إزالة أقفال كل الحزم التي تبدأ بـ"yast2	Zypper rl 'yast2*'
عرض الأقفال الموجودة	Zypper locksاو zypper ll

** الخدمات

* التحقق من الإعتماديات : zypper veاو zypper verify .
* ملحوظة : هذا مفيد في حالة إنهيار النظام .
* تثبيت الحزم الموصى بها من قبل حزم أخرى مثبتة : zypper inrاو zypper install-new-recommends .

** ترقية التوزيعة

* ترقية التوزيعة : zypper dupاو zypper dist-upgrade .

** إستخدام Zypper في التطبيقات و السكربتات

* الوضع غير التفاعلي ... zypper –non-interactive مثال : zypper –non-interactive patch تخطى كل الرقع التي تطلب تفاعل من المستخدم .
* عدم التحقق من الـ GPG : zypper –-no-gpg-checks
* الموافقة تلقائيا على الرخص : . zypper –auto-agree-with-licenses
* عرض مخرجات الأمر بطريقة هادئة سهلة القراءة : . zypper –-quiet
* عرض ناتج الأمر بنمط XML : zypper –xmlout

الحلقة الاولى
عمل تطبيقات الويب WSGI

بقلم : مؤيد السعدي

1.1 - تمهيد عن السلسلة

بمناسبة إطلاق مشروع إطار الويب المتواضع الخاص بمشروع ثواب والذي يحمل اسم الصحابي عكاشة بن محصن (رضي الله عنه) قررت أن أكتب سلسلة سنتعلم فيها ماذا توفر لنا لغة البرمجة بايثون من طرق لعمل تطبيقات ويب. في هذه الحلقة سأشرح إطار الويب الماورائي meta web framework للغة بايثون واسمه WSGI. وفي الحلقة القادمة (إن شاء الله) سنتحدث عن إطار الويب "عكاشة".

1.2 ما قبل WSGI

1.2.1 هل تذكرون CGI

لا أعرف إن كان هناك من لا زال يستعملها لكن لا بأس من التعرف عليها. الطريقة التقليدية في عمل تطبيقات الويب التفاعلية هي أن يقوم الخادم بتنفيذ برنامج غالبا سكربت موجود في (cgi-bin) يمرر له ما يُريد منه عبر متغيرات البيئة environment variables ثم يأخذ المخرجات التي تكون وفق معايير HTTP أي أنها ترويسات بينها سطر جديد ثم بعد الترويسات سطر جديد آخر ثم يبدأ محتوى الرد. مثلا لنكتب تطبيق CGI بلغة bash كما يلي:

```
#! /bin/bash
echo 'Content-Type: text/plain'
echo ''
echo 'Hello, world!'
```

هذه الطريقة تسمى CGI أي Common Gateway Interface وهي لا تتعلق بأي لغة معينة حيث تكون بأي لغة يمكن للخادم تنفيذها.

أما التفاعلية فتأتي من أننا يمكننا أن نطبع ما نشاء مثل الوقت أو نعمل cat لمحتويات ملف أو نتحدث مع قاعدة بيانات أو نعالج ما تم تمريره لنا عبر متغيرات البيئة مثل المتغير QUERY_STRING يحمل الطلب (الذي يكون بعد علامة الاستفهام "؟" التي بعد اسم الملف) مثلا إن كان اسم الملف هو test.sh وتم زيارة http://localhost/cgi-bin/test.sh?text=foo&id=2 فإن قيمة ذاك المتغير هي text=foo&id=2

عيوب هذه الطريقة أنه مع كل طلب جديد يصل للخادم يتم تحميل برنامج وتنفيذه يعني لو قام ١٠ أشخاص بزيارة موقعك فإن البرنامج سيتم تنفيذه ١٠ مرات في كل مرة تشعيب (fork.)

ومن عيوبها كثرة الأعمال المملة التي عليك القيام بها لعمل أشياء بسيطة وكما نعلم كلما زاد التعقيد قل الأمن (مثلا قد تنس عمل escape لبعض المتغيرات قبل عرضها)

1.2.2 mod_python و fast_cgi وغيرها من الطرق

لتوفير تحميل مفسر اللغة التي كتب لها السكربت فإن استخدام مفسر واحد مضمن داخل الخادم يكون حل أفضل وهذا هو ما يقوم به mod_python حيث أنه جزء من خادم أباتشي وكلما يصل طلب يقوم mod_python الموجود مسبقا في الذاكرة باستلامه ومعالجته. لكن يجب كتابة التطبيق ليستعمل mod_python وليس cgi.

وهناك العديد من الطرق لكتابة تطبيقات الويب منها أن تكتب خادم ويب خاص بك حتى يكون لك كامل التحكم وتتجنب تحميل ثم تشغيل تطبيق الويب مع كل طلب (لأن تطبيق الويب هو نفسه الخادم) ومما يسهل عليك ذلك SimpleHTTPServer[1] أو خادم الويب في paste.httpserver[2] أو في twisted-web[3] وفي جميع الأحوال يمكن

تشغيل هذا الخادم داخليا على منفذ عالي مثل ٨٠٨٠ ثم الطلب من خادم أباتشي أن يكون مجرد وكيل يمرر الطلبات من العالم الخارجي له وبالعكس.

أما fast_cgi فهي بروتوكول متعدد اللغات يعمل في الوسط بين تطبيق cgi العادي (يعني لا داع لإعادة كتابة تطبيق cgi) والخادم فيوفر عدد مرات تشغيل التطبيق من جديد.

كل تلك الحلول تتطلب كتابة التطبيق بطرق مختلفه لكل حل (ف cgi أو fast_cgi تختلف تماما عن mod_python وهما يختلفان عن twisted-python وهكذا). لنأخذ مثلا على تطبيق واحد مثل ويكي اسمها MionMion[4] التي ما هي إلا تطبيق ويب بلغة بايثون فلو كنت أنت مطورها فلعلك تسأل نفسك وفق أي من هذه الحلول ستكتبها ؟ وإن قررت دعم أكثر من واحد فإن تبين وجود مشكلة أمنية في أحد التنفيذات هل تنعكس على الاخرى أم لا ؟ هل ستعيد كتابة الكثير من الأجزاء المملة من البرنامج ؟

طيب إن كنت ستصمم إطار ويب مثل Django أو Pylons فأي من هذه الأسالب ستدعم ؟ ماذا ستقول للمستخدمين الذين لا يدعم خادمهم إلا واحدة دون الأخريات من تلك الطرق.

1.3 إطار الويب الماورائي ويز-جي WSGI

1.3.1 ما هو ويز-جي WSGI

ويز-جي اختصار لكلمة Web Server Gateway Interface وهي تلفظ بأكثر من طريقة منها wiz-gee

وهي ليست إطار ويب بل هي الطريقة التي يجب أن تكتب بها أطر الويب وتطبيقاته في بايثون. لهذا يسمى meta framework أي أنه إن كتبت التطبيق بهذه الطريقة فإنه سيعمل بسهولة على أي حل تختاره.

مثلا في جافا هناك tomcat وهناك jetty لكن من يعمل تطبيقات ويب بلغة جافا لا يهتم هل الخادم يعتمد على tomcat أم jetty لأن التنقل بينهما شفاف فهما يحدثان ما يسمى في عالم جافا باسم خويدمات جافا .Java Servlet وهذا هو تماما ما تعمله WSGI في عالم بايثون.

وقبل أن تسأل لا يوجد حزمة اسمها WSGI ولا تحتاج عمل import لأي شيء. بل هي الطريقة التي يجب أن يكتب بها برنامجك.

1.3.2 كيف يعمل ؟

إن أردت كتابة تطبيق ويب متوافق مع ويز-جي فإن تطبيق الويب الخاص بك إما أن يكون مجرد دالة واحدة أو أن يكون صنف class لكنه قابل للاستدعاء عبر الدالة call

هذه الدالة تستلم معاملين هما environ و start_response أولهما عبارة عن قاموس بايثوني (أو ما يسمى associated array في لغات أخرى) عناصره متغيرات البيئة وما يقابلها من قيم ومنها ٥

* - SCRIPT_NAME وهو المسار من أول URI لتطبيق الويب ويجوز أن يكون خاليا
* - PATH_INFO وهي المسار الذي يلي التطبيق في URI مثلا إن كان الرابط هو http://localhost/app/view/file.txt يمكن أن يكون app هو SCRIPT_NAME و الباقي يعني / view/file.txt هو الملف المطلوب PATH_INFO
* - QUERY_STRING الاستعلام المطلوب وهو الذي يكون بعد علامة الاستفهام في الروابط عندما يكون الطلب من نوع GET مثلا http://www.google.com/search?q=ojuba
* - REMOTE_ADDR وهو عنوان IP للطرف البعيد الذي أرسل الطلب يفيد لعمل limits مثلا
* - REQUEST_METHOD وهي طريقة الطلب وهي غالبا إما GET أو POST

أما المعامل الثاني فهو عبارة عن دالة يجب أن يستدعيها التطبيق قبل أن يبدأ إرسال النتائج. فائدة استدعاء هذه الدالة

هي تمكين الخادم إرسال ترويسة http وتستدعى هذه الدالة مرة واحدة في كل استجابة ويمرر لها معاملين

* الأول هو status وهو سلسلة نصية تحتوي نوع الرد غالبا "OK ٢٠٠"[6]
* الثاني هو قائمة عناصرها ازواج من الترويسة وقيمتها وأهمها طبعا "content-type" وقيمتها يمكن أن تكون "text/html; charset=utf-8" مثلا.

أما الصفحة نفسها تولد عبر إعادة قائمة بأجزاء الصفحة مثلا

```
 return ['Hello, world!']
```

ويجوز أن نعيد أي كائن قابل للتسلسل iteratable سواءً قائمة أو مرتب أو حتى قطعة نصية واحدة أو generator أو حتى كائن ملف.

1.3.3 تطبيق ويب بسيط على شكل دالة

لنفرض أننا نريد عمل تطبيق ويب بسيط يعرض اسم الصفحة التي طلبت منه ويعرض الوقت منذ بزوغ يونكس بالثواني.

```
import time
def application(environ, start_response):
 start_response("200 OK",[('content-type', 'text/html;
charset=utf-8')]);
  return ['''<html><body>
<h1>Hello!</h1>
<p>you have requested the page [%s]</p>
<p>it's [%d] since Unix epoch</p>
</body></html>''' % (environ['PATH_INFO'],time.time())]
```

لاحظ أن تطبيق الويب السابق لا يستعمل أي وحدات إضافية ولاحظ كم هو بسيط. سنبين الآن كيف يمكننا تشغيل تطبيق الويب هذا على أنواع مختلفة من الخوادم

* تطبيق cgi عبر خادم apache
* عبر google web app engine
* عبر mod_wsgi في خادم apache
* إطلاق خادم خاص بالتطبيق عبر paste

1.3.4 تطبيق ويب بسيط على شكل صنف class

يمكن عمل تطبيق الويب على شكل class وتنتقل مهمة الدالة إلى الطريقة call داخل الصنف هكذا

```
import time
class MyWebApp:
  def __init__(self):
    pass

  def __call__(self, environ, start_response):
   start_response("200 OK",[('content-type', 'text/html;
charset=utf-8')]);
    return ['''<html><body>
<h1>Hello!</h1>
<p>you have requested the page [%s]</p>
<p>it's [%d] since Unix epoch</p>
</body></html>''' % (environ['PATH_INFO'],time.time())]

application=MyWebApp()
```

لاحظ أننا قمنا بعمل كائن فرد **instance** من ذاك الصنف وفائدة هذا تمرير أي معاملات لازمة لإنشاء التطبيق كأن تكون معاملات معايرة أو إعدادات معينة وهي التي ستمرر للطريقة **init**

1.3.5 تشغيل التطبيق السابق على CGI

من السهل جدا تحويل أي تطبيق ويز-جي ليعمل كتطبيق **CGI** عبر الكود العياري الذي تم التقدم به في **PEP 0333** وهي تعني **Python Enhancement Proposals** هذا الكود تجده على الرابط

* **http://www.python.org/dev/peps/pep-0333/#the-server-gateway-side**

حيث نستدعي الدالة **run_with_cgi** المعرفة في ذاك الملف (سمه مثلا **wsgi2cgi**) ونمرر لها تطبيقنا هكذا

```
from myTestWebApp import application
from wsgi2cgi import run_with_cgi
run_with_cgi(application)
```

نضع هذا في مجلد **cgi-bin** في الخادم ونعطيه صلاحيات التنفيذ وغير ذلك من الإعدادات التقليدية كأي تطبيق **CGI** تقليدي. وبالمناسبة نفس الشيء يكون في **fast_cgi**

وطبعا كود البرنامج الذي يشغله لا يحتاج الكثير من الشرح فدالة **start_response** ترسل الترويسات بعد تنسيقها ثم ترسل علامة سطر جديد

```
for header in response_headers:
  sys.stdout.write('%s: %s\r\n' % header)
sys.stdout.write('\r\n')
```

ثم تسير على البيانات التي أعادها التطبيق وتكتبها إلى المخرجات ثم **sys.stdout.flush**

1.3.6 تشغيل التطبيق السابق على غوغل

طبعا لا داع أن نُذكّر أن غوغل تحجب بعض خدماتها بطريقة انتقائية (أي ولا تحجب خدمات أخرى) عن بعض الدول العربية التي تعتبرها الولايات المتحدة "دولا مارقة" فيما يفترض أنه تطبيق لقوانين التصدير الأمريكية [7] . ومن هذا البعض خدمة غوغل كود وخدمة استضافة تطبيقات بايثون **google app engine** وهي خدمة مفيدة جدا حيث يمكن استضافة أي تطبيق مكتوب بلغة بايثون أو جافا. سنشرحها للفائدة فقط مع تقديم النصيحة بعدم استعمالها للأسباب المذكورة أعلاه.

الطريقة هي بكل بساطة عبر استدعاء **run_wsgi_app** وتمرير تطبيق الويب خاصتنا له هكذا

```
from myTestWebApp import application
from google.appengine.ext.webapp.util import run_wsgi_app

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

للمزيد انظر

http://code.google.com/appengine/docs/python/tools/webapp/utilmodule.html#run_wsgi_app

1.3.7 تشغيل التطبيق السابق mod_wsgi

يوجد وحدة لخادم أباتشي اسمها mod_wsgi تسمح بتشغيل تطبيقات ويز-جي بكل سهولة انظر
http://code.google.com/p/modwsgi/
كل ما عليك هو أن تعمل ملف إعدادات (في فيدورا اعمل ملف جديد داخل المجلد / etc/httpd/conf.d/) يحتوي السطر التالي

```
WSGIScriptAlias /myapp /path/to/myapp.wsgi
```

حيث myapp / الأولى هي المسار الذي سيتم تشغيل التطبيق عند زيارته. الثاني / path/to/myapp.wsgi هو ملف مكتوب بلغة بايثون (يعامل وكأنه وحدة module) يحتوي تعريف لكائن أو متغير اسمه application وطبعا يمكن أن يكون هو نفسه الوحدة السابقة لأننا عرفنا فيه متغير/كائن بهذا الاسم. إن لم تكن تملك صلاحيات تثبيت الوحدات في النظام يمكنك وضعها في أي مكان وإضافتها للمسار الخاص ببايثون هكذا

```
import sys
sys.path.insert(0,'/home/omar/my-python-libs/')
from myTestWebApp import application
```

إن كان الصنف بحاجة لتمرير معاملات معينة يمكنك استيراد النصف ثم استهلال كائن منه هكذا

```
import sys
sys.path.insert(0,'/home/omar/my-python-libs/')
from myTestWebApp import MyWebApp
application=MyWebApp(arg1,arg2)
```

وهنا نحن لا نستدعي الدالة MyWebApp بل ننشئ كائن من نوع MyWebApp ونسمي الكائن application سيتم استدعاء application مع كل طلب يصل إلى الخادم.

1.3.8 إطلاق خادم خاص بالتطبيق عبر paste

إن تجربة تطبيق الويب من خلال خادم ضخم مثل أباتشي أمر مزعج تخيل نفسك تعطل خدمة الويب وتعيد تشغيل أباتشي لمجرد تعطيل سطر صغير لذلك فالأسهل هو استخدم خادم paste المدمج أثناء تطوير البرنامج. أضف الأسطر التالية إلى نهاية تطبيق ويز-جي

```
if __name__ == '__main__':
    from paste import httpserver
    httpserver.serve(application, host='127.0.0.1', port='8080')
```

السطر الأول يعني أن ما بعده ينفذ فقط إن تم تنفيذ الملف ولا ينفذ إذا كنا نستعمله كوحدة module السطر الذي يليه تحضر خادم ويب من حزمة python-paste (موجودة في مستودع فيدورا بهذا الاسم) وتشغل تطبيقنا application على هذا الخادم على المنفذ ٨٠٨٠ (يمكنك تغير الرقم لأي رقم غير محجوز فوق ١٠٢٤) ويستمع للطلبات من الجهاز نفسه. إن جعلتها ٠,٠,٠,٠ فإنه سيستجيب لكل الطلبات عبر الشبكة المحلية أو الخارجية.

بعد إضافة تلك الأسطر فقط قم بتنفيذ ملف بايثون الذي يحتوي على خادم paste ثم افتح المتصفح على المنفذ المحدد في المثال /http://127.0.0.1:8080
عندما تريد تغيير أي شيء يمكنك إيقاف البرنامج من سطر الأوامر بالضغط على CTRL+C
* للمزيد انظر http://pythonpaste.org/modules/httpserver.html

<-->

1انظر http://docs.python.org/library/simplehttpserver.html وانظر من تطبيقاته pydoc حيث يمكن تشغيله كخادم بالأمر pydoc -p 8080
2انظر /http://pythonpaste.org
3انظر http://twistedmatrix.com/trac/wiki/Documentation
4انظر /http://moinmo.in
5للمزيد منها انظر http://www.python.org/dev/peps/pep-0333/#environ-variables
6للمزيد انظر http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html#sec6.1
7للمزيد انظر مدونة محمد بشير النّعيمي

تقنية تبادل الملفات الحرّة

Magnet

بقلم : جمعان الزهراني

-اخترت أن أتحدث عن هذه التقنية لعدة أسباب :

* تطبيق مبدأ نشر العلم والمعرفة.
* تقنية رابط Magnet هي مشروع حر.
* عدم وجود مصادر عربية تشرح هذه التقنية.
* رد جميل وعرفان لمجتمع لينوكس المعطاء العربي.

حينما نتحدث عن عالم الشبكات ، إحدى المميزات التي تكون في أي شبكة هي إمكانية المشاركة . فتخيلوا معي عندما نتحدث عن شبكة عالمية مثل الإنترنت، طبعاً ستكون هناك مشاركة للملفات والبيانات على نطاق ضخم وواسع.



المقالة ستستعرض إحدى تقنيات المشاركة والتبادل وهي من نوع (نظير إلى نظير) أو بالإنجليزية (peer-to-peer) وهي تقنية (Magnet URL Link رابط المغناطيس).

- ما هي تقنية رابط Magnet؟

هي بيئة مفتوحة لمشاركة البيانات (ملفات أو مجلدات) محددة في رابط URL محدّد بين أجهزة الكمبيوتر (النظائر) عبر شبكة الانترنت.

مثال لرابط Magnet :

magnet:?xt=urn:sha1:YNCKHTQCWBTRNJIV4WNAE52SJUQCZO5C

-ما هي إستخدامات هذه التقنية ؟

بالإمكان تنزيل أي ملف بمعرفة رابط Magnet الخاص بهذا الملف. ويتم استخدام هذه التقنية في مواقع المشاركة (نظير إلى نظير) على وجه الخصوص لأسباب عدة سنتطرق لها لاحقاً.

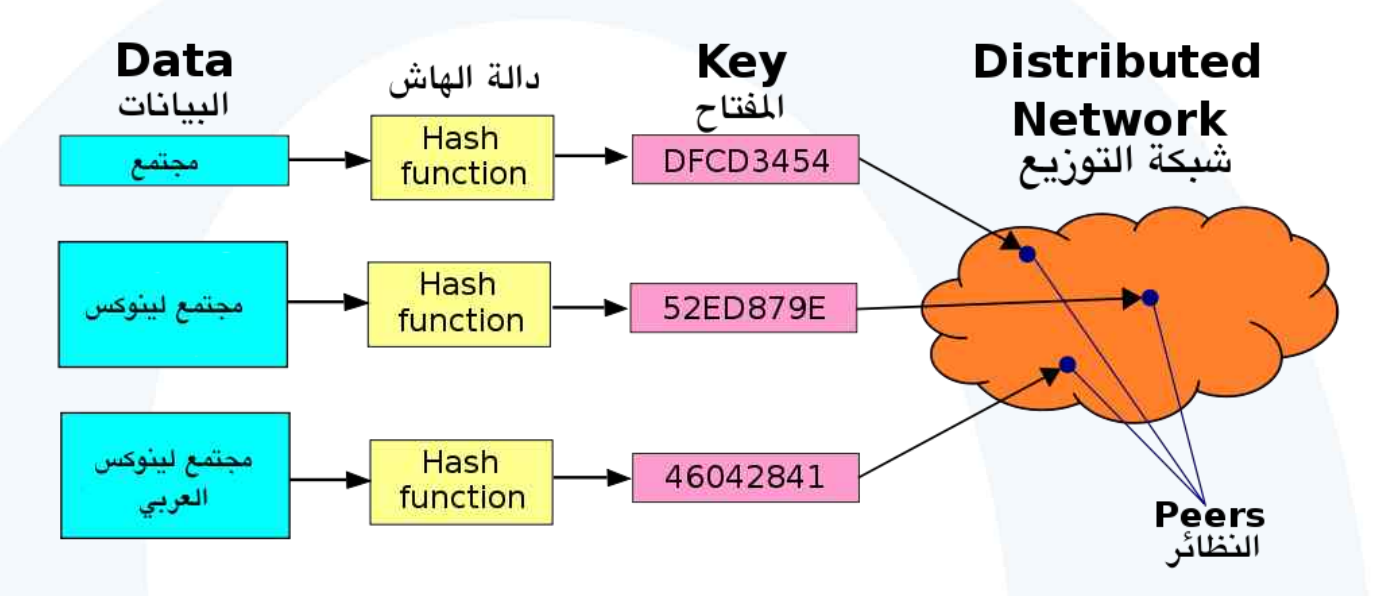
-ما هي آلية عمل هذه التقنية ؟

#عند طلب رابط Magnet يتم البحث عن مصدر (نظير) عن طريق شفرة مفتاحية للبيانات

#يتم الاتصال بشبكة توزيع تسمى Distributed Hash Tables حيث تنظم البيانات بين النظائر.

#بعد ذلك تبدأ عملية التنزيل والرفع بين النظائر

صورة توضح آلية العمل بشكل أدق :



-ما أسباب ظهور روابط Magnet URL؟

1 . ضمان مشاركة البيانات بين النظائر لمدة لا نهائية.
حيث إستمرارية البيانات تعتمد على النظائر وتواجدهم. ولن يكون هناك أي حاجة لملفات من نوع (torrent.)
فقط ما تحتاجه هو رابط Magnet .

2 . ضمان مشاركة البيانات بين النظائر دون وسيط .
حيث في تقنية التورنت لابد من وجود مركز ينظم النظائر ويسمى التراكر ، ولكن في تقنية Magnetلا يوجد ما يسمى تراكر حيث يكون توزيع البيانات – كما ذكرنا - عبر شبكة التوزيع DHT .
(لن تفكر في مشاكل التراكر من تعليق أو عطل بعد الآن)

سلبية واحدة واجهتني وهي البطء في بداية التحميل (حيث خلال هذه المدة يتم الاتصال عبر شبكة التوزيع وجلب النظائر) ولكن بعد جلب واستيراد كافة البيانات لن يكون هناك أي بطء إطلاقاً.

-كيف يمكنني إستخدام روابط Magnet؟

لإستخدام التقنية تحتاج لبرنامج مشاركة يحوي هاتين التقنيتين :

١. تقنية شبكة التوزيع DHT
٢. إمكانية قراءة روابط من نوع Magnet

يا رجل أريد استخدامها في نظام جنو/ لينوكس ؟
حسناً ، برامج المشاركة التي تدعم هاتين التقنتين ومتواجدة في أغلب مخازن التوزيعات .

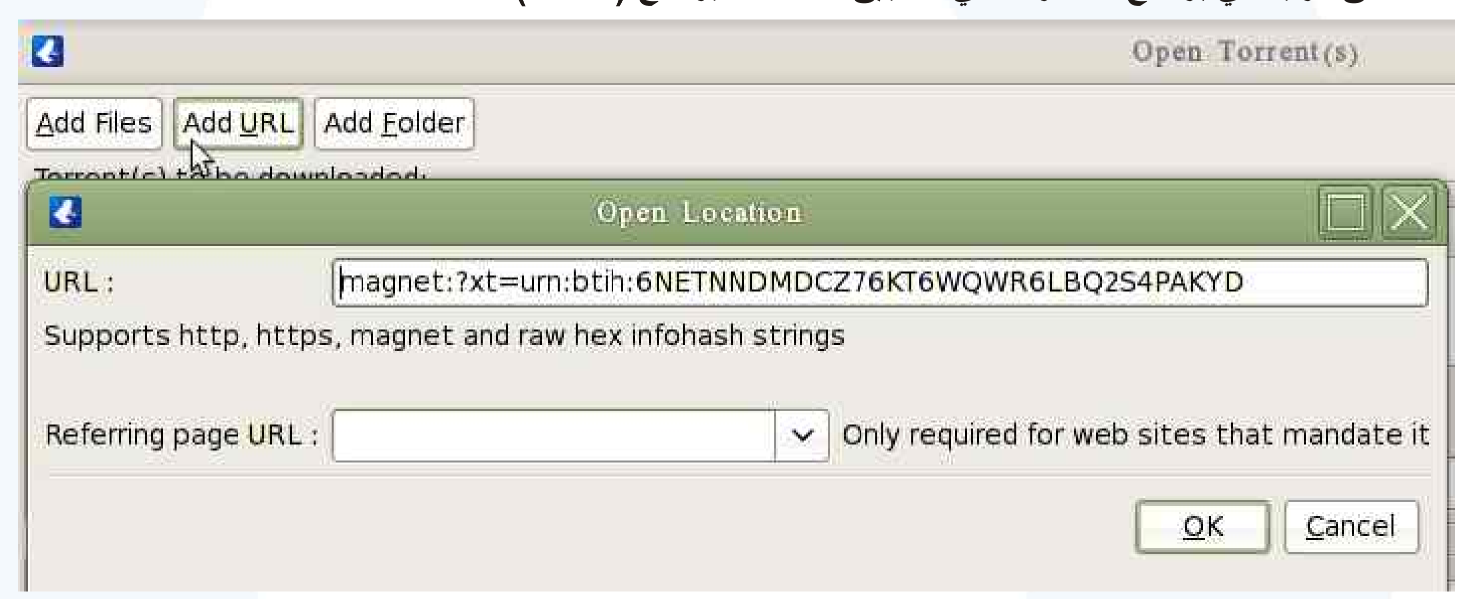
١. برنامج (Transmission 1.8الإصدار التجريبي يدعم هذه التقنية فقط).
٢. برنامج Vuzuمعروف بإسم (Azureus)وهو بالمناسبة أول برنامج يدعم هذه التقنية.
٣. برنامج Deluge

الطريقة :

١. نسخ رابط Magnet للبيانات المراد تنزيلها .



٢. لصق الرابط في برنامج المشاركة في التطبيق استخدمت برنامج (Vuzu)



النتيجة :

Icon	Name	Subscribe	Size	Qualit	Info	Progress	Seeds
	Ahmed_Alajmi_(MP3_Quran)		1.77 GB			0.2% (347.9 kB/s) ETA 1h 47m	20

Name: Ahmed_Alajmi_(MP3_Quran) 0.2% D: 347.9 kB/: U: 0 B/s ETA: 1h 47m

- مراجع ساعدتني في المقالة وروابط مفيدة :

* موقع المشروع : http://magnet-uri.sourceforge.net
* مقالة عن التقنية في موسوعة ويكيبيديا http://en.wikipedia.org/wiki/Magnet_URI_scheme
* مقالة توضح الفروق بين برامج المشاركة في ويكيبيديا http://en.wikipedia.org/wiki/Comparison_of_BitTorrent_clients
* مقالة عن DHT في ويكيبيديا : http://en.wikipedia.org/wiki/Distributed_hash_table
* محرك بحث عن روابط Magnet (يعيبه أن النتائج غير حديثة) http://bitzi.com/
* التراكر العربي المفتوح : http://www.arabp2p.com

* خاتمة :

أحمد الله أن ساعدني على إنهاء هذه المقالة المصغرة . واعذروني لو بدر مني خطأ أو نقص.
بقي أن أذكر أن هذه العمل لوجه الله وهو حر ومفتوح المصدر .

والله ولي التوفيق .

مثلث أمن المعلومات
و المصادر المفتوحة

بقلم : أبو عمــــر

مثلث أمن المعلومات :

أمن المعلومات هي الشغل الشاغل للعديد من المنظمات و الهيئات و المؤسسات سواء علي النطاق الحكومي العام او الخاص ، و بذلك أصبح أمن المعلومات يفرض نفسه علي الساحه العالمية و العربية ، و من هنا وجب ان نقوم بتوعية المستخدم العربي بسبل الوصول إلي تحقيق عوامل أمن المعلومات ، و عوامل أمن المعلومات تنحصر في ثلاثة عناصر هامة لا يتم تحقيق الأمان للمعلومة في ظل غياب أي منها و هي (السريه – السلامه – التوفر)

السريه :

و هي الحفاظ علي المعلومة و عدم وصول اي شخص غير مصرح له بالوصول اليها و يتم من خلال تأمين الشبكة و عمل سياسه امنيه للمنظومة الخاصه بالعمل .

السلامه :

و هي سلامة المعلومة من الفقدان أو التلف و يتم ذلك من خلال عمل نسخة أحتياطية من المعلومات و البيانات الموجودة لدينا للرجوع اليها حين الحاجه لذلك.ئ

التوفر :

و هو التأكد من توفر المعلومة و استمرار التواصل بين المستخدم و المعلومة و عدم انقطاع عن المستخدم و يتم ذلك من خلال عمل التأمين اللازم للشبكة الداخلية و الخارجية للمنظومة .

المصدر المفتوح Open Source :

قامت فكرة المصدر المفتوح علي أن تتاح الشفرة التي يتم كتابة البرامج او التطبيقات بها للمستخدم بحيث ان يتمكن من التعديل عليها او اعادة صياغتها او استخدامها دون اي قيود و بحريه تامة و من أهم تطبيقات المصادر المفتوح نظام التشغيل – (- GNU/Linuxالمجاني) المنافس لنظام – – Windowsالمغلق المصدر و الذي يتراوح سعرة من ١٠٠$ الي ٢٠٠$ في الأسواق .

انترنت بلا وعي :

أعتمدت الحكومات العربيه اليه عمل جديد تحت مصطلح مصطلح جديد و هو " الحكومة الألكترونية " و لم يتم التحضير الجيد لهذة النقلة و بناء علي عدم التحضير حدث الكثير من الأختراقات و في ظل غياب الوعي الأمني نجد الكثير من المشاكل التي تواجه هذه الحكومات بشكل مستمر و في هذه المقالة نضع الحل بكل سهولة و يسر و سوف نشرح المخاطر الأمنية من أستخدام مصدر مغلق داخل الجهات المؤسسات و المنظمات و الشركات و الهيئات الحكومية – هذا ليس تقليل من شأن مجهودات الحكومات العربية بل هو محاولة لأظهار بعض الحقاق حول كوارث تهدد أمن المجتمعات و الحكومات العربيه فهناك من يستخدم windows 98 الي الان داخل بعض المنشأت التابعه للحكومة و هناك من يستخدم الي الان الطرق اليدوية في الأرشفه و حفظ المعلومات التي تنتهي مع (أول ماس كهربائي يحدث داخل المنشأه)
.

المصادر المغلقه و هدم مثلث أمن المعلومات :

إعترفت ميكروسوفت من قبل بأن الأنظمة Windows 95/98 كانت تحتوي علي برامج تقوم بأرسال تقارير عن البرامج التي يقوم المستخدم بتنصيبها علي الجهاز و بررت ذلك العمل لحفظ حقوقها من خلال معرفة البرامج المقرصة من البرامج الأصلية التي يستخدمها المستخدم – و في هذا العمل خرق لأول قاعدة من قواعد مثلث أمن المعلومات و هي السرية – فأين هي السرية و هناك من يعلم بكل ما تقوم به علي الحاسب الخاص بك .

الثانية أمر يخص - Windows NT Service Pack 5 الجدير بالذكر ان هذه النسخه من ويندوز اعتمدت عليها الكثير من الشركات و المؤسسات و المنظمات و الهيئات الحكومية – أعلن - – Cryptonym of Morrisville Dr. Andrew Fernandez في ولاية كارولينا الشمالية بالولايات المتحدة و ذلك عندما وجد ان بعض مهندسي مايكروسوفت المهملين لم يزل رموزالتصحيح Debugging من نسخة خدمة النظام SP 4 لنظام تشغيل ويندوز NT ليجد أن هذه الأكواد تشير في أحد مفاتيحها إلي وكالة الأمن القوم الأمريكي NSAKEY و قد أعلن البروفيسور عن كشفه في مؤتمر التشفير لمتقدم advanced Cryptography المقام في سانتا باربارا.

و من قبله أعلن Dr Nicko van Someren البريطاني الجنسية أنه وجد ان احد سواقات البرامج المحملة تلقائيا علي أنظمة تشغيل ويندوز و المسمي ADVAPI.DLL و الذي يتحكم في بعض أهم خصائص سرية النظام و خصوصا المرتبطة بمتصفح انترنت اكسبلورر يحوي أكواد مجهولة تستطيع أن تشغل برامج خاصة من خلال الشبكات عند ادخال الرموز المناسبة لها إلا أنه لم يمكن التعرف عي نوعية البرامج التي تشغلها مثل هذه الأكواد و ما هي وظيفتها و هو ما رفضت مايكروسوفت التعليق عليه في حينه. و قد أدي الكشف الأول إلي اصدار الحكومة الألمانية قرارا بمنع إستخدام أي برامج تنتجها شركة مايكروسوفت علي اجهزة الحكومة.

و بربط الكشف الثاني مع الأول يمكن التعرف علي الجهة التي تم ادخال أكواد التجسس لصالحها و هي وكالة الأمن القومي الأمريكي.

أدعوكم لتصفح موقع الشركة صاحبة الأكتشاف - www.cryptonym.com

بالطبع هنا أيضاً نجد هدم كامل لرأس مثلث أمن المعلومات ، هذا بالأضافة الي الكثير من الثغرات الموجودة في النظام نفسه و ما علي المستخدم سوي أن ينتظر الشركة المصنعه للبرنامج ان تقوم بمدة بالتحديثات التي عادة لا تظهر سوي بعد ايام و قد تصل إلي أسابيع من تاريخ أكتشاف الثغرة . في ظل أستخدامك لمصدر مغلق لن تتمكن من تحليل النظام و معرفة ما يحتويه و إن تم أكتشاف أي خطأ برمجي يستوجب التعديل علي المصدر فلن تتمكن من ذلك بالأضافة إلي أستخدامك إلي كراكات لتشغيل بعض البرامج التي تحتاجها اثناء عملك اليومي و في العادة فأن ٩٠% من الكراكات و الباتشات الموجودة علي شبكة الأنترنت بعد فحصها تأكد وجود ملفات تجسس بها و هنا أختفي عنصر السلامة أحدي عناصر مثلث أمن المعلومات.

المصدر المفتوح هو المنقذ:

بعد ما قدمناه من استعراض لمخاطر استخدامنا المصادر المغلقه نأتي الي عرض ايجابيات أستخدام المصادر المفتوحة في تحقيق عوامل أمن المعلومات ، أولاً لك الحرية الكاملة أن تقوم بالتعديل او التطوير للنظام و ايضاً لا يتم فرض اي مواد محدده الاستخدام لك فأنت من يقرر ماذا يستخدم و ماذا يلغي ، هذا بالأضافة إلي ان جميع البرامج التي تعمل علي انظمة التشغيل مفتوحة المصدر هي أيضاً برامج حرة لا تطلب منك كراك او باتش – و هنا حققنا عنصر السلامة للمعلومات و الحفاظ علي بيئة المعلومة – ايضاً تجد عنصر اخر متوفر و هو عنصر السرية فالأنظمة مفتوحة المصدر تمكنك من الأطلاع علي الشفرة المصدرية للبرنامج الذي تعمل عليه و من هنا لا يتم كتابة أي اكواد تجسس او اكواد تتبع لعمليات و استخدامك للحاسب الخاص بك .

لا ننكر وجود مشاكل و فيروسات داخل الأنظمة مفتوحة المصدر لكن الفارق بينها و بين المصادر المغلقه هو وجود ملايين من المطورين حول العالم يقدمون الدعم الكامل للمصادر المفتوحة فأستخدامك للمصادر المفتوحة يجعلك في أمان دائم و يحقق لك عنصرين مهمين من عناصر مثلث أمن المعلومات – السرية ، السلامة – تبقي لدينا ركن التوفر و هي مسئله قائمة علي المستخدم نفسه و طرق اتصاله بالمعلومة الخاصة به و يتيح لك ايضاً المصدر المفتوح جزء من تحقيق هذا العنصر و هو سهولة الوصول لكافة بياناتك و معلوماتك و تطبيقاتك و ذلك من خلال توفير برامج مجانية و مفتوحة المصدر توفر لك كل ما تشاء من الاستخدامات المتنوعة للحاسب .

نستعرض معاً بعض المؤسسات و الشركات التي تعتمد علي المصادر المفتوحة :

وزارة الدفاع الأمريكية و المخابرات الأمريكية – و هما من أكبر الأجهزة الأمنية علي مستوي العالم و يحتاجون الي درجة عاليه من السرية و استخدامهم للمصادر الحرة يدل علي توفر ذلك .

وزارة الطاقة الأمريكية و وكالة الأمن القومي الأمريكية و هما لا يقلان عن السابقتين شئ و ايضاً شركات أمازون لبيع الكتب يستخدمون داخل منظومتهم و خوادمهم انظمة المصادر المفتوحة ووكالة رويترز و هي إحدي أكبر الوكالات الأخبارية في العالم تستخدم المصادر المفتوحة و محرك البحث جوجل ايضاً يعتمد أعتماد كلي علي المصادر المفتوحة ، و هناك الكثير من الشركات يعمل بأنظمة المصادر المفتوحة لتحقيق عوامل مثلث أمن المعلومات.

تم بحمد الله و كل عام و انتم بخير



بقلم : أمين روخ

مقدمة:

Vimعبارة عن برنامج لتحرير النصوص بمختلف أنواعها، مايميزه هو سهولته وخصائصه الكثيرة؛ سهولته تكمن في سرعة تعلمه وإنجاز الأعمال به، وخصائصه تمكن من تسريع وتيرة العمل ورفع مستوى الإنتاجية به.

هناك العديد من مستخدمي Vimممن يكتفي بتعلم الأساسيات فقط والقليل منهم يصل لدرجة إحترافه والتعمق في المزايا التي يوفرها، البرنامج ليس فقط للمبرمجين بل يمكن إستخدامه من طرف المستخدمين العاديين وفي أي مكان من أجل تحرير النصوص أيا كان نوعها، البرنامج يعمل في ويندوز وماك وسولاريس ويأتي إفتراضيا في نظام لينوكس.

الموضوعنا لن يتناول الأساسيات بل الإستخدام المتقدم للبرنامج.

البداية:

vimيحتوي على ثلاث أنواع من الوضعيات الرئيسية: العادية، الإدخال، المرئية ((normal, insert, visual، فقط للتذكير بالأوامر الأساسية للبرنامج إن كنت جديدا أو نسيتها، لتشغيله من الطرفية أكتب vimثم إسم الملف:

```
vim filename $
```

سيفتح الأمر البرنامج للتحرير الملف filenameفي المسار الإفتراضي للبرنامج، للخروج أكتب:

```
q:
```

أو للخروج من دون حفظ التغييرات:

```
!q:
```

في حالة التحرير يمكنك الإنتقال في مختلف إتجاهات الملف:

h اليسار
j الأسفل
kالأعلى
l اليمين

تم إختيار هذه التشكيلة من المفاتيح بدل الأسهم العادية (ضمن لوحة المفاتيح) لأنها تقع في صف واحد وستحفظ الكثير من الوقت أثناء التحرير.
لكتابة النصوص إضغط على المفتاح iستنتقل إلى وضعية الإدخال، من المفروض يسظهر أسفل يسار شاشة الطرفية مايلي:

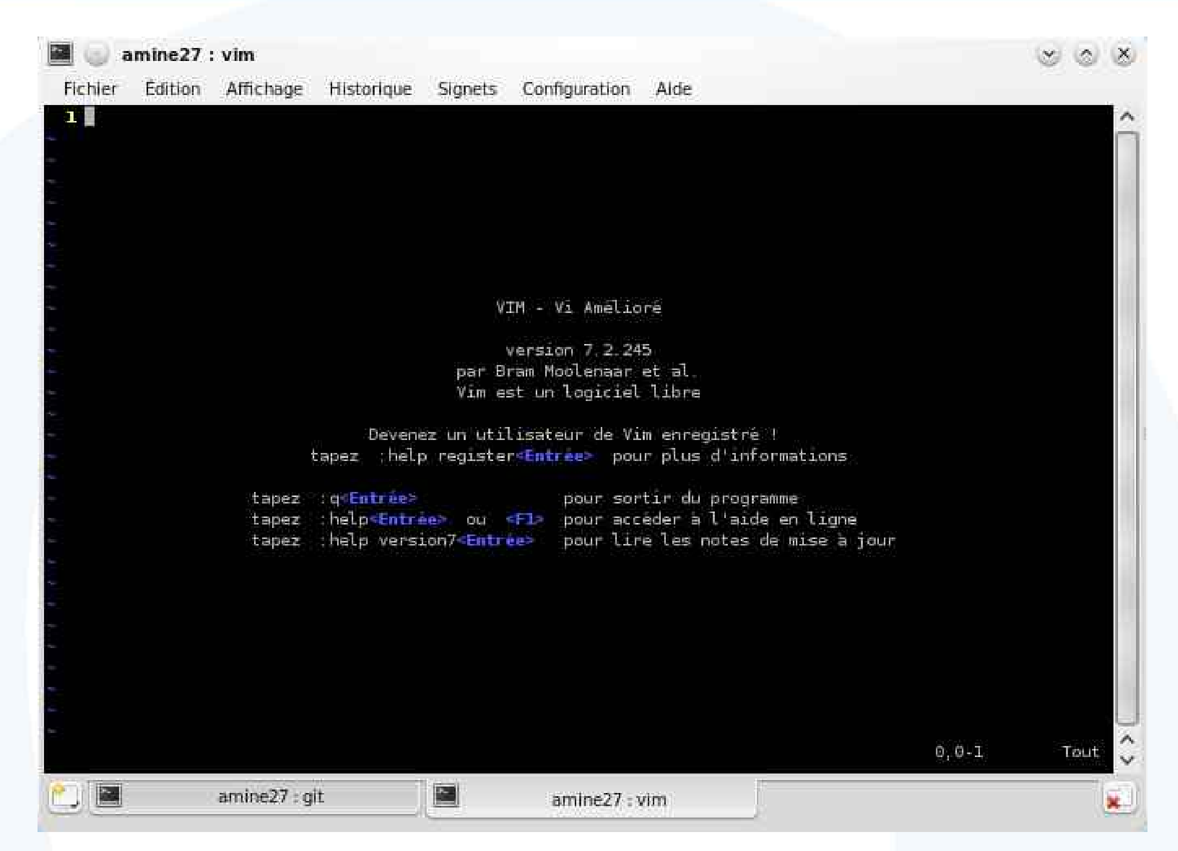
```
- - INSERT - -
```

يظهر لك البرنامج وضعيته الحالية، بعد إنتهائك من الكتابة إضغط على <ESC>سترجع الآن إلى الوضعية العادية (الإفتراضية)، يمكنك الإعتياد عليها بعد بعض التمرين والممارسة، للحفظ:

```
w:
```

أو للحفظ والخروج معا:

```
wq:
```



وضعية الإدخال:

معظم الأوقات يكون vim في وضعية الإدخال عند الكتابة. لأنها أهم مرحلة لدى المستخدمين يوفر vim خصائص كثيرة لإدخال الأحرف والكلمات في أي سطر كنت والإنتقال في مختلف الإتجاهات ببعض ضغطات على لوحة المفاتيح.

من الوضعية العادية إضغط:

i للإدخال قبل المؤشر
a للإدخال بعد المؤشر (الإضافة)
shift+I للإدخال عند بداية السطر
shift+a للإدخال عند نهاية السطر

كما يلاحظ هي متشابهة لتسهيل حفظها على المستخدمين، لإنشاء الأسطر:

o إنشاء سطر بعد السطر الحالي ووضع المؤشر عليه
shift+o إنشاء سطر قبل السطر الحالي ووضع المؤشر عليه

دائما من الوضعية العادية، للقيام ببعض التعديلات على الأحرف:

r لإستبدال الحرف عند المؤشر بحرف آخر
shift+r للإنتقال لوضعية إستبدال الأحرف في جميع النص

لحذف الأحرف:

x لحذف الحرف الحالي
shift+x لحدف الحرف قبل المؤشر

الوضعية المرئية:

التحديد في vim يتم عن طريق الفأرة، لكن يفضل تحديد النصوص عن طريق الوضعية المرئية. من خلال الوضعية العادية إضغط على v، شاهد أسفل يسار الشاشة:

```
- - VISUAL - -
```

التحرك من خلال المفاتيح h، j، k، l سيحدد النص، يمكنك حذف الجزء المحدد بـ x، نسخ ولصق بـ p، y، بعد الإنتهاء يمكنك العودة إلى الوضيعية العادية بـ .Esc

التراجع والعودة:

قد تقع في أخطاء أثناء الكتابة، كيف ترجع للخلف ؟ Ctrl+Z؟ لا. في الوضعية العادية ببساطة إضغط على u، بالنسبة للتقدم فيتم عبر Ctrl+R

يمكنك التراجع إلى الوراء حتى تصل إلى نقطة بداية العمل فبرنامج vim يقوم بحفظ جميع التعديلات، حتى أنه يمكنك التراجع للخلف وفق وقت معين:

```
:earlier 5m
```

أو

```
:earlier 1h
```

للعودة خمس دقائق أو ساعة للخلف.

النسخ، القص، اللصق:

ي vim النص المحدد يمكن نسخه بـ y كما أشرنا سابقا، أما أمر اللصق فهو p، جميع مايتم حذفه أو نسخه يذهب إلى مسجل خاص بالبرنامج لذا يمكن العودة إلى الخلف دائما، الأوامر الرئيسية :

y نسخ النص المحدد
yy نسخ السطر الحالي
Y نسخ السطر الحالي
d قص النص المحدد
dd قص السطر الحالي
D قص النص من وضعية المؤشر الحالية إلى نهاية السطر
p لصق النص

يمكنك الإستفادة من خاصية المسجلات كالتالي:

fy :f” نسخ النص المحدد ولصقه في المسجل
fd :f” حذف النص المحدد ولصقه في المسجل
fp :f” لصق محتويات المسجل

للإطلاع على محتويات المسجل أكتب:

```
:reg[f]
```

حافظة vim مستقلة عن نظام التشغيل، ويقوم بجلب النص من حافظة النظام إلى مسجل رمزه '+'، وبستخدمه أيضا لوضع نص من vim إلى نظام تشغيل.

الذاكرة المساعدة:

أوامر الذاكرة المساعدة أو Mnemonics عبارة عن أوامر يمكن تذكرها من خلال المهام التي تقوم بها، لهذا تم فصلها عن الأوامر العادية، منها على سبيل المثال:

ctx – change till x إحذف من هنا إلى غاية الحرف س
dtx – delete till x

تسمى هذه الخاصية بـ text-object للمزيد من التفاصيل حولها أكتب في :vim

```
:help text-objects
```

بعض الأوامر الأخرى ضمن هذه الخاصية:

daw إحذف كلمة
dw إحذف منها إلى نهاية الكلمة
daW إحذف كلمة أو الأحرف المخصصة
das إحذف الجملة الحالية
daB إحذف البلوك من الكود

هذه الأوامر تستعمل في الوضعية العادية لكن يمكن أيضا إستخدامها في الوضعية المرئية، لتحديد جملة إضغط على as أو كلمة بـ aw

ترقيم الأوامر:

جميع الأوامر يمكن تقديم كتابتها برقم لتحديد عدد مرات تكرارها، مثال:

٦j إنزل أسفلا ستة أسطر
dd 10 إحذف عشرة أسطر

هذه الخاصية مفيدة للغاية لتسريع تحرير النصوص.

التصفح بطريقة متقدمة:

الطريقة البسيطة:

يوفر لك vimطرقا عدة لبدأ التحرير مباشرة عند كلمة أو حرف معين ببضع نقرات فقط، نبدأ أولا بالتحرك وفق الكلمات:

w تقدم بكلمة واحدة للأمام وضع مؤشر الفأرة أمامها
e ضع المؤشر عند نهاية الكلمة الحالية
b ضع المؤشر عند بداية الكلمة السابقة

الأهم طبعا نهاية وبداية الأسطر، للتموضع وفق ماتريد:

٠ ضع المؤشر عند بداية السطر
^ ضع المؤشر عند أول حرف ضمن السطر
$ ضع المؤشر عند عند نهاية السطر

للإنتقال إلى الصفحة السابقة أو الموالية (PageUp/Down) يستعمل vim هذه الأزرار:

Ctrl+f إذهب إلى الصفحة التالية
Ctrl+b إذهب إلى الصفحة السابقة
Ctrl+u إذهب إلى منتصف الصفحة التالية
Ctrl+d إذهب إلى منتصف الصفحة السابقة

الأمر :g

الأمر g يستعمل للتنقل ضمن الملف :(goto):

١G إذهب إلى بداية الملف
٠G إذهب إلى نهاية الملف
gD إذهب إلى بداية الجملة الحالية بطريقة ذكية (مثلا تعريف بداية الدالة في البرمجة)

طبعا الأمر g لايستعمل فقط للتنقل بل يتعداه إلى أوامر أخرى منها:

إفتح إسم الملف عند المؤشر gf
إطبع مايقابل الحرف التالي في جدول ascii من خلال: ga
أدخل vim في سبات مدة ثانية (يمكن سبقها بعدد ١٠ مثلا لـ ١٠ ثوان) gs

للمزيد من التفاصيل شغل :help g

أمر التعليم:

أمر التعليم (mark) في vim مفيد للغاية، حيث يمكنك من تعليم سطر ما ثم العودة إليه فيمابعد:

تعليم الموضع التالي بالحرف a مثلا ma
العودة إلى العلامة `a
ضع المؤشر في بداية السطر المعلم 'a
لمشاهدة العلامات :marks

البحث:

البحث في vim سهل، فقط أكتب / الكلمة المراد البحث عنها ثم Enter للتأكيد، إضغط على n للنتيجة الموالية، و N أو Shift+n للنتيجة السابقة، أيضا تستعمل # و * لتحديد الكلمة ضمن نتائج البحث للأمام أو الخلف:

البحث /foobar<Enter>
النتيجة التالية n
النتيجة السابقة N
الكلمة عند المؤشر للوراء #
الكلمة تحت المؤشر للأمام *

التصفح للمبرمجين:

للمبرمجين هناك أمر مهم للإنتقال من بداية البلوك إلى نهايته، هو % علامات البلوك هي {} و () و [] و <> وغيره، أيضا الأمر يستعمل g للإنتقال إلى تعريف الدالة:

إذهب إلى تعريف الدالة gD
الإنتقال إلى اسم الملف تحت المؤشر (مفيدة للإنتقال إلى header مثلا) gF

الطيات:

ستعمل الطيات (Folds) لتنظيم الأكواد في vim إلى مجموعات، مثل أكواد LaTex وغيره، أفضل طريقة لفهم الطيات يتم عن طريق مثال طبعا، افتح ملف نصي يحتوي على نص معتبر (أكثر من صفحة في شاشة المحرر) بواسطة vim، ضع مؤشر الفرأة في أي مكان تريد وفي الوضعية العادية أكتب:

```
zf3j
```

بعد الضغط على زسيقوم vim بطي الصفحة لإظهار ٣ أسطر التالية وإخفاء الأسطر في الأعلى، تم ذلك بفضل الأمر zf أما ٣j فتعني الإنتقال ثلاثة أسطر للأسفل

يمكن إستعمال الأمر أيضا في الوضعية المرئية، فقط حدد جزء من النص ثم إضغط على zf ستختفي تلك الأسطر، يمكن أيضا تحديد الأسطر المراد طيها بمجال، مثلا من السطر ٣ إلى ١١٠:

```
:-3, 110 fold
```

إن كنت تحرر ملفا برمجيا (سي مثلا) لطي بلود من الأسطر البرمجية ضع المؤشر عند } ثم أكتب:

```
zfa}
```

الأمر السابق يمكن تطبيقه بـ:

```
za%
```

يمكن دمج هذه الخاصية بخاصية البحث، لطي النص من الحرف الحالي إلى الكلمة المراد البحث عنها، طبق مايلي:

```
z/SearchString
```

طبعا الطي نصف المهمة، فلا بد من إرجاع النص إلى حالته في مرحلة من مراحل، قم بوضع المؤشر عند مكان الطي ثم أكتب:

```
zo
```

لإعادة طيه بعد الإنتهاد من التعديل:

```
zc
```

للإنتقال بين مختلف الطيات الموجودة في النص:

zj للطية التالية
zk للطية السابقة

الطيات المتداخلة:

يدعم vimتداخل الطيات فيمابينها، إذا ضغطت على zoستفتح فقط طية واحدة والباقي سيبقى مغلقا، للتعامل مع الطيات المتداخلة نستعمل:

zO إفتح جميل الطيات المتداخلة ضمن الطية الحالية
zr إفتح فقط المستويات العلوية من الطيات (الأولى)
zR إفتح جميع الطيات في جميع المستويات
zm أغلق جميع المستويات الدنيا من الطيات
zM أغلق جميع الطيات في جميع المستويات
zE إحذف جميع الطيات

عند حذف الطيات لايمكن التراجع للوراء في .vimعند البحث يتفصح vim الطيات ويفتحها إن وجد نتيجة لكنه لايعيد إغلاق الطية، zmتقوم بهذه المهمة سريعا

عند إغلاق الملف جميع الطيات ستذهب، لذا يجب حفظها قبل الإغلاق، عن طريق الأمر:

```
:mkview
```

وإسترجاعها بواسطة:

```
:loadview
```

أو يمكن جعل العملية آلية بإضافة هذان السطرين إلى ملف .vimrc:

```
au BufWinLeave * mkview
au BufWinEnter * silent loadview
```

amine27 : vim

Fichier Edition Affichage Historique Signets Configuration Aide

```
1 #include "declaration.h"
2
3 void permuter(element *x, element *y)
4 +--  7 lignes : {-----------------------------------------------------------
11
12 void triRapide(list *aList, int g, int d)
13 {
14     int i, j, pivot;
15
16     if (g < d) // cas d'arrèt pour la récursivité
17     {
18         i = g;
19         j = d+1;
20         pivot = (*aList)[g].mat;
21
22         do
23 +-----  12 lignes : {--------------------------------------------------------
35         while (i < j);
36
37         permuter(&((*aList)[g]), &((*aList)[j]));
38
39         //affichage(aList, N); // N nombre d'enregistrement
40         triRapide(aList, g, j-1); // trie partie1
41         triRapide(aList, j+1, d); // trie partie2
42     }
43 }
~
~
~
~
                                                       23,1          Tout
```

amine27 : vim   amine27 : vim

المرشح:

يمكن من خلال vimتنفيذ أوامر الشل بكتابة:

```
:! shell-command
```

على طريقة غنو المعروفة، وهنا نستفيد من خاصية المرشحات (Filters)التي يوفرها vim، حيث يمكن دفع النص الحالي إلى إحدى أوامر الشل (sortمثلا) ثم أخذ النتيجة وإرجاعها إلى vim، الأمر التالي:

```
:%!gpg
```

يقوم بتشفير النص المفتوح في vim(والمعبر عنه بـ %) عن طريق البرنامج gpg، مثال آخر:

```
:%tr a-z n-za-m
```

سيقوم هذا الأمر بتطبيق خوارزمية ROT13على ملفنا.

تصنيف مشروع:

بواسطة خاصية المرشحات يمكن تصنيف (compile)مشروعنا البرمجي عن طريق vim مباشرة:

```
:!gcc -Wall -g foo.c -o bar
```

أو لفحص الصياغة فقط:

```
:!gcc -fsyntax-only %
```

vimيحتوي أيضا على أداة make، لتفعيلها:

```
:set makeprg=gcc\ -g\ %
```

make: لإستدعائها

كما تقبل أوامر make العادية مثل:

```
:make all
:make install
```

إذا حدثت أخطاء أثناء التصنيف يمكن مشاهدتها عبر:

cc: لمشاهدة الخطأ الحالي
cn: لمشاهدة الخطأ التالي
cN: لمشاهدة الخطأ السابق
copen: لمشاهدة جميع الأخطاء في شاشة منفصلة

هذه الأوامر ستسرع كثيرا من وتيرة البرمجة.

التعامل مع الملفات المتعددة:

أحيانا كثيرة، في البرمجة خاصة، نفتح أكثر من ملف في وقت واحد، لتسهيل ذلك يستخدم vim خاصية split (للتفاصيل help vsplit:) لكن منذ النسخة السابعة ظهرت خاصية أكثر عملية هي خاصية التبويب. لفتح ملف في تبويب جديد:

```
:tabnew file.txt
```

للإنتقال بين مختلف التبويبات:

```
:tabnext (tabn)
:tabprevious (tabp)
```

ثم إغلاق التبويب الحالي:

```
:tabclose (tabc)
```

طريقة أخرى للتعامل مع الملفات المتعددة هي إستخدام خاصية المخزن (Buffer)، كل ملف له مخزن خاص به عند فتحه إلى غاية إغلاقه، لفتح ملف:

```
:e file.txt
```

للتعامل مع المخزنات:

bfirst: إذهب إلى المخزن الأول
bn: إذهب إلى المخزن التالي
bp: إذهب إلى المخزن السابق
blast: إذهب إلى المخزن الأخير
ls: أظهر جميع المخزنات المفتوحة
buf file.txt: إنتقل إلى الملف file.txt في المخزن

المفيد مع المخزنات هو أنه يمكن تطبيق نفس الأمر على جميع المخزنات المفتوحة دفعة واحدة:

```
:bufdo cmd
```

مثلا للبحث في جميع الملفات:

```
:bufdo /searchString/
```

أو للبحث ثم التعويض:

```
:bufdo %s/oldString/newString/ge
```

يشار أنه يوجد أيضا الأمر tabdo في خاصية التبويب يقوم بمهام مشابهة للأخير.

amine27 : vim

Fichier Édition Affichage Historique Signets Configuration Aide

quicksort.c ~/D/I/C/t/declaration.h ~/D/I/C/t/main.c

```c
#ifndef DECLARATION_H
#define DECLARATION_H

#include <stdio.h>
#include <stdlib.h>

#define MAX_TAB 1000
#define MAX 32765

typedef struct
{
    int mat;
        char nom[20];
        char prenom[20];
} element;

typedef element list[MAX_TAB];

#include "lecture.c"
#include "afficher.c"
#include "lectureFichier.c"
#include "ecritureFichier.c"
#include "fusionsort.c"
#include "heapsort.c"
#include "insertsort.c"
#include "insertsort-bin.c"
#include "quicksort.c"
#include "quicksort-med.c"
```

1,1 Haut

amine27 : bash amine27 : vim

تصفح الملفات:

عند فتح مسار من خلال vimعن طريق :e pathأو عند تشغلية، يمكن حينها تصفح الملفات والتعديل عليها أيضا. بعد الأوامر المهمة:

أنشئ مجلد **d**
إحذف مجلد **D**
أعد تسمية **R**

يمكن أيضا الإنتقال بين المجلدات عن طريق الأمر :cd

```
:cd path/to/dir
```

أو معرفة المسار الحالي بـ pwd

التعيين والإختصار:

يعتبر تعيين الأوامر (map)ميزة مهمة ضمن أي محرر، حيث يختصر الوقت في كتابة الأوامر الطويلة ومتكررة الإستعمال، vimيوفر خاصية تعيين الأوامر ضمن الوضعية العادية (nmap)ووضعية الإدخال ((imap، الأمر يكون على الشكل:

```
:map <keystrokes> command-to-be-executed
```

مثال:

```
:nmap \cc !gcc\ -g\ %
```

بعد تطبيقه، في الوضعية العادية وعند كتابة \ccيتم تصنيف الملف المفتوح بواسطة gcc، هذا لإختصار الأوامر، إختصار الجمل يكون على الشكل:

```
:iab LAC Linux Arabic Community
```

عند كتابة LACمتبوعة بمسافة يتم تحويلها آليا إلى **Linux Arabic Community**، الحرف i يدل على وضعية الإدخال أي يتم تطبيق الإختصار فقط في وضعية الإدخال، لحذف التعيين أو الإختصار نطبق:

```
:unmap \cc
:unab LAC
```

لحفظ هذه التعيينات والإختصارات عند كل تشغيل للبرنامج، أضف الأسطر السابقة إلى ملف .vimrc

التكملة التلقائية:

خاصية التكملة التلقائية متوفرة في vim، عبر:

Ctrl+n أكمل (ستظهر قائمة منبثقة)
Ctrl+x f أكمل إسم الملف
Ctrl+x i أكمل إسم الملف الرأسي

مراجعة التهيئة:

الإصدار السابع من vim يدعم خاصية مراجعة التهيئة (SpellCheck) تلقائيا، لتفعيلها:

```
:set spell
```

للإنتقال بين الأخطاء أكتب]s و [s للأمام والخلف، للتصحيح أكتب =z ستظهر لك قائمة بالكلمات المحتملة، أدخل رقم الكلمة الصحيحة ثم أكد، لتغيير لغة المراجعة أكتب الأمر:

```
:setlocal spell spelllang=en_us
```

مع تغيير en_us إلى مايوافق لغة نصك، في الأخير لتعطيل الأمر:

```
:set nospell
```

بعض الخصائص المهمة:

نستعرض مجموعة من أوامر vim التي قد تكون مفيدة للبعض، الصيغة العامة لتفعيل أو تعطيل الخاصية تكون:

```
:set option
:set nooption
```

بعضها:

```
:set option=value
```

لمعرفة حالة الخاصية (مفعلة/معطلة):

```
:set option?
```

وهذه بعضها:

الإزاحة الآلية للمؤشر عند الأسطر الجديدة autoindent
الإزاحة الذكية، أفضل من الإزاحة الآلية لأنها تعرف مكان الرجوع إلى بداية السطر من عدمه (مفيدة أثناء البرمجة) smartident
إبراز نتائج البحث بلون مختلف hlsearch
البحث التزادي، تحديث نتائج البحث أثناء الكتابة الكلمة المراد البحث عنها incsearch
عند الضغط على زر Tab يتم تغيير التبويب بمسافات محددة expandtab
إبراز بداية ونهاية البلوك البرمجي ({} مثلا) showmatch
تغيير صيغة الملف الحالي إلى القيمة الجديدة filetype=value

يكون تطبيق الأوامر السابقة من الشكل:

```
:set ft=html
```

لتغيير صيغة الملف الحالي إلى ملف html، أو:

```
:set hlsearch
:set nohlsearch
```

لتفعيل أو تعطيل خاصية إبراز نتائج البحث، لتفعيل تلوين النصوص حسب نوعها وإظهار أرقام الأسطر (مفيدة للمبرمجين) طبق:

```
:syntax on
:set number
```

لإظهار الأمر الحالي أسفل الشاشة:

```
:set showcmd
```

يمكن أيضا إستعمال الفأرة مع vim، لتفعيلها:

```
:set mouse=a
```

أوامر متفرقة:

لجلب السطر الموالي إلى نهاية السطر الحالي Shift+j
أظهر وثائق المساعدة (man page) للكلمة المؤشر عليها Shift+k
لعكس حجم الحرف (Upper إلى Lower والعكس) ~
أعد تنفيذ آخر أمر .
أضف القيمة ١ إلى الرقم المؤشر عليه Ctrl+A
أنقص القيمة ١ من الرقم المؤشر عليه Ctrl+X

إضافات مفيدة:

الإضافات تعطي لـ vim قوة أكبر، فهي تسهل المهام كثيرا وتقدم خدمات رائعة للمستخدمين، نذكر منها:

تظهر قائمة للدوال المتوفرة مثل أي بيئة تطوير taglist explorer
تفعيل خاصية تطابق التعابير النمطية (Regular expressions) عن طريق % matchit
تشفير وفك تشفير الملفات بمفتاح خاص gpg
لإظهار التقويم مع إمكانية إظافة الأحداث وفق الأيام clendar
لإظهار جميع التبويبات والمخزنات المفتوحة أفقيا أسفل شاشة البرنامج minibufexplorer
لعبة المكعبات الشهيرة حاضرة أيضا tetris

يوجد غيرها العديد على موقع vim بالتأكيد.

في النهاية:

طبعا يبقى هذا الشرح مقتضبا نظرا لما يحتويه vim من خصائص وميزات كثيرة يصعب التطرق إليها جميعا، لكن يجب في البداية التعود على البرنامج وطريقة عمله مع حفظ الأساسيات من الأوامر، بعدها ستجد نفسك تقضي أعمالك في مدة زمنية قصيرة مقارنة مع المحررات الأخرى، وهنا تكمن قوة vim، إحرص أن تبق دائما في الوضعية العادية؛ بعد إنتهائك من التحرير إرجع فورا إلى تلك الوضعية لأنها الأساس، تذكر أيضا أن الأوامر يمكن أن تسبق برقم لتحديد عدد مرات تكرارها (3j(هذه أيضا خاصية جد مفيدة، وأخيرا لاتنس تخصيص ملف vimrc. بمايناسبك من الأوامر المساعدة، وعند أية إستفسار أكتب help: فستجد فيه معلومات قيمة.

بالتوفيق والسلام

System Documentation
CFG2HTML

بقلم : حمدي هشام مرشد

السلام عليكم ورحمه الله وبركاته

يستخدم الكثير من مدراء الأنظمة أكثر من أداه لمراقبة النظام وإستخراج تقارير ومشاهدة مواصفات السيرفر من معالج ورام وهارديسك وغيرها ...
على سبيل المثال اذكر بعض الأوامر لبعض هذه العمليات

```
cat /proc/cpuinfo
```

سيظهر لك معلومات المعالج وال core وعددها وعده تفصيلات لكل منها

```
last| grep boot
```

سيأتي لك بآخر عمليات إعادة التشغيل للسيرفر

```
/usr/bin/lshal
```

سيظهر لك تفاصيل قطع السيرفر سواء .. pci or usbوبتفاصيل مفصلة بشكل دقيق ..

```
hdparm -t -T /dev/hda
```

سيظهر لك سرعة القرص في النقل وال bufferوغيرها..

....

ماذا لو قلنا أن هناك أداه واحدة تقوم بإنشاء تقارير كاملة عن كل هذا واكثر بكثير ....
خصوصا من يهتمون بال Documentation
سواء مدراء سيرفرات او مهتمون بالصيانة او الشبكات او غيرهم ..

ربما من الصعب جدا أن أحصي هذه التفاصيل عن هذه الأداه .. ولكن سأحاول ان أذكر بعضها

CPU and Model info
هو نفسه الأمر الذي ذكرته بالأعلى
OS, Kernel version
نظام التشغيل والنواه بالتفصيل
Uptime
فترة تشغيل الخادم
Top load processes
أكثر العمليات المسببة لل **load**
Top memory consuming processes
أكثر عمليات المستهلكة للذاكرة
Services Startup
الخدمات التي تعمل عند تشغيل الجهاز
Cron
العمليات المجدولة واوقاتها وبتفصيل لكل مستخدم
Hardware(Physical Memory List of Devices (lshal) USB devices PCI devices ACPI IoPorts Interrupts SCSI Components SCSI Devices DMA Devices Serial

Devices Serial ttyS0 Serial ttyS1 IDE Disks CDROM Drive)
تفاصيل القطع (العتاد) كل على حده وبشكل رائع
Packages installed
الحزم المنصبة ..
FileSystem
تفاصيل انظمة الملفات المستخدمه
Disk Partitions
اقسام الهارديسك وايضا بتفصيل
LAN Interfaces
كروت الشبكة وتفاصيلها
Route Configuration Files
اعدادات التوجيه في السيرفر

والقائمة طويلة جدا ....

الميزة الأكثر جمالا وفائدة هي انه يكتب لك الامر المستخدم في اظهار كل معلومة ....
الأداه متوفرة لأغلب أنظمة لينكس وبأكثر من طريقة للتنصيب حسب النظام أو حسب ما تريد أنت سواء حزم deb او rpm أو حتى ال source

طريقة التنصيب:

```
cd /root/
wget http://www.cfg2html.com/cfg2html-linux-1.64-
20091215_all.zip
unzip cfg2html-linux-1.64-20091215_all.zip
rpm -ivh cfg2html-linux-1.64-4.noarch.rpm
```

ولمستخدمي حزم deb آخر امر استبدله بـ

```
dpkg -I cfg2html-linux_1.64-4_all.deb
```

```
File  Edit  View  Terminal  Help

100%[=======================================>] 341,054      191K/s   in 1.7s

2009-12-27 12:36:28 (191 KB/s) - `cfg2html-linux-1.64-20091215_all.zip' saved [3
41054/341054]

root@server2 [~]# unzip cfg2html-linux-1.64-20091215_all.zip
Archive:  cfg2html-linux-1.64-20091215_all.zip
  inflating: cfg2html-linux-1.64-4.noarch.rpm
  inflating: cfg2html-linux_1.64-4.dsc
  inflating: cfg2html-linux_1.64-4.tar.gz
  inflating: cfg2html-linux_1.64-4_all.deb
  inflating: cfg2html-linux_1.64-4_i386.changes
  inflating: README_cfg2html.txt
  inflating: descript.ion
 extracting: Packages.gz
root@server2 [~]# rpm -ivh cfg2html-linux
cfg2html-linux-1.64-20091215_all.zip  cfg2html-linux_1.64-4_i386.changes
cfg2html-linux_1.64-4_all.deb         cfg2html-linux-1.64-4.noarch.rpm
cfg2html-linux_1.64-4.dsc             cfg2html-linux_1.64-4.tar.gz
root@server2 [~]# rpm -ivh cfg2html-linux-1.64-4.noarch.rpm
Preparing...                ########################################### [100%]
   1:cfg2html-linux         ########################################### [100%]
root@server2 [~]#
```

```
pico /etc/cfg2html/systeminfo
```

طريقة التشغيل ...لاظهار النتائج

```
cfg2html
```

او

```
cfg2html-linux
```

الإختلاف فقط في إظهار خلفيه :)

ناتج الامر

```
root@server2 [~]# cfg2html-linux
```

http://come.to/cfg2html...

```
Starting            cfg2html-linux version 1.64-2009-09-23
Path to Cfg2Html    /usr/bin/cfg2html-linux
HTML Output File    ./server2.al-ra3ed.com.html
Text Output File    ./server2.al-ra3ed.com.txt
Partitions          ./server2.al-ra3ed.com.partitions.save
Errors logged to    ./server2.al-ra3ed.com.err
Started at          2009-12-27 12:39:12
WARNING             USE AT YOUR OWN RISK!!! :-))
```

http://come.to/cfg2html...

```
Collecting:  Linux System CentOS release 5.4 (Final)
...............................
Collecting:  Cron and At  ..............
Collecting:  Hardware  .....................
Collecting:  Software  ....
Collecting:  Filesystems, Dump- and Swapconfiguration
...........
Collecting:  Multipath Configuration  ........
Collecting:  LVM  ........
Collecting:  Network Settings  .................................
Collecting:  Kernel, Modules and Libraries  ..............
Collecting:  System Enhancements  .
Collecting:  Applications and Subsystems  ........
Collecting:  Local files  .
```

http://come.to/cfg2html...

```
Creating:    JPG/GIFs
x - cfg2html_back.jpg [non-ascii]
x - profbull.gif [non-ascii]
```

```
File  Edit  View  Terminal  Help
root@server2 [~]# cfg2html-linux


--=[ http://come.to/cfg2html ]=----------------------------------------------
Starting          cfg2html-linux version 1.64-2009-09-23
Path to Cfg2Html  /usr/bin/cfg2html-linux
HTML Output File  ./server2.al-ra3ed.com.html
Text Output File  ./server2.al-ra3ed.com.txt
Partitions        ./server2.al-ra3ed.com.partitions.save
Errors logged to  ./server2.al-ra3ed.com.err
Started at        2009-12-27 12:39:12
WARNING           USE AT YOUR OWN RISK!!! :-))

--=[ http://come.to/cfg2html ]=----------------------------------------------

Collecting:  Linux System CentOS release 5.4 (Final)  ...............................
Collecting:  Cron and At  .............
Collecting:  Hardware  ....................
Collecting:  Software  ....
Collecting:  Filesystems, Dump- and Swapconfiguration  ...........
Collecting:  Multipath Configuration  ........
Collecting:  LVM  ........
Collecting:  Network Settings  ................................
Collecting:  Kernel, Modules and Libraries  ..............
Collecting:  System Enhancements  .
Collecting:  Applications and Subsystems  ........
Collecting:  Local files  .

--=[ http://come.to/cfg2html ]=----------------------------------------------
Creating:    JPG/GIFs
x - cfg2html_back.jpg [non-ascii]
x - profbull.gif [non-ascii]
root@server2 [~]#
```

هنا يكون السكربت قام بانشاء التقرير الكامل لكل ما ذكرت ..

server.test.com.html
server.test.com.txt
server.test.com.partitions.save
server.test.com.err

الاول هو الملف الرئيسي للتقرير ويمكنك نقله لاي موقع تريد على السيرفر وعرضه عبر المتصفح

```
cp server.test.com.html /home/test/public_html/
```

وربما يكون السيرفر امامك فيمكنك عرضه بشكل عادي ..

اما باقي الملفات فالثاني نسخة نصية من التقرير

والثالث ملف خاص بتقسيمات القرص ونفس المحتوى موجود بالتقرير فليس له فائدة ..
الاخير خاص بالاخطاء خلال جمع المعلومات ...

صور من النتائج ...

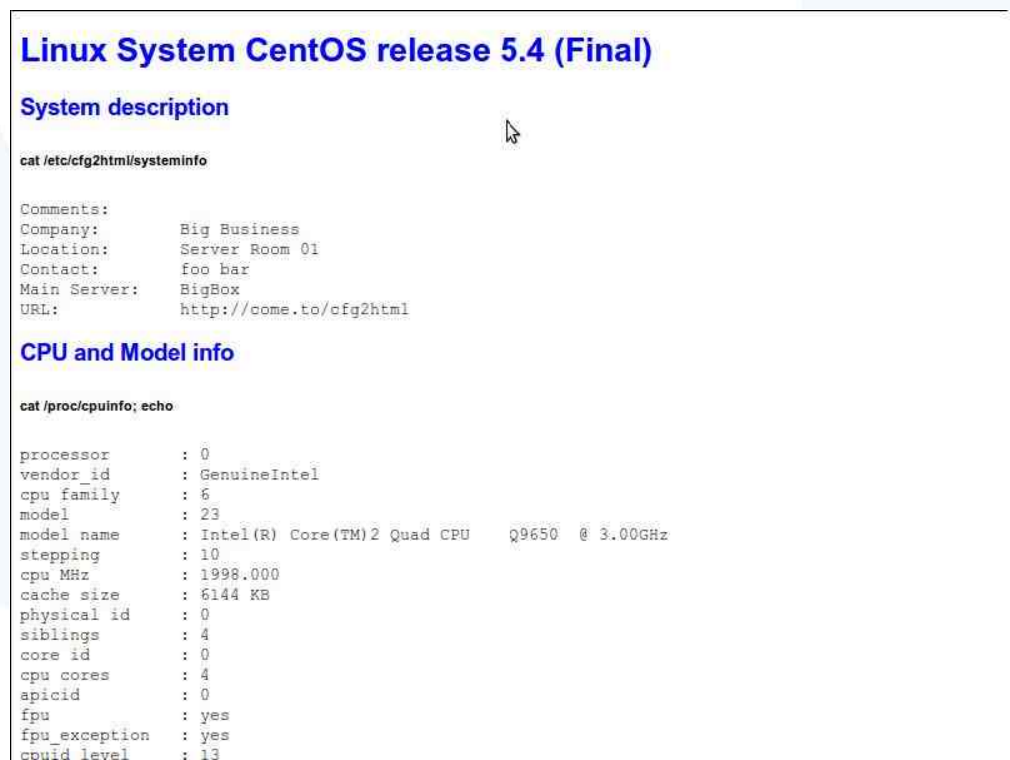

server2.al-ra3ed.com - System

Created 2009-12-27 12:50:22 with cfg2html-linux version 1.64-2009-09-23

Linux server2.al-ra3ed.com 2.6.18-164.6.1.el5.centos.plus #1 SMP Wed Nov 4 09:31:39 EST 2009 x86_64 x86_64 x86_64 GNU/Linux

Contents

- Linux System CentOS release 5.4 (Final)
  - System description
  - CPU and Model info
  - uname & hostname
  - Host alias
  - OS, Kernel version
  - Linux Standard Base Version
  - OS Specific Release Information (/etc/redhat-release)
  - POSIX Standards/Settings
  - locale specific information
  - System ulimit
  - VM-Statistics 1 10
  - VM-Statistics (Summary)
  - MP-Statistics
  - IO-Statistics
  - Uptime
  - Active Process Overview
  - Top load processes
  - Top memory consuming processes
  - reboots
  - Alias
  - inittab


Linux System CentOS release 5.4 (Final)

System description

cat /etc/cfg2html/systeminfo

```
Comments:
Company:        Big Business
Location:       Server Room 01
Contact:        foo bar
Main Server:    BigBox
URL:            http://come.to/cfg2html
```

CPU and Model info

cat /proc/cpuinfo; echo

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 23
model name      : Intel(R) Core(TM)2 Quad CPU    Q9650  @ 3.00GHz
stepping        : 10
cpu MHz         : 1998.000
cache size      : 6144 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 4
apicid          : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
```

إلى هنا .. انتهى دوري في عرض هذه الاداة الرائعة والمميزة والبسيطة حتى والكبيرة في ناتجها ..

...وما تبقى فهو عليك لتكتشف أهميتها لك كمدير نظام او دعم فني ..

ملاحظة يمكنك زيارة موقع السكربت للإطلاع على آخر التحديثات

http://www.cfg2html.com

BitDefender Antivirus Scanner
for Unices

بقلم : مصطفى رفاعي

بالطبع لا أحتاج لمضاد فيروسات و لا أعتقد أن هناك مستخدم جنو/لينكس يحتاجه .. و لكن أحيانا قد يحتاج البعض مضاد فيروسات من أجل فحص سواقة فلاش مهيئة بنظام FAT32 او NTFS او قرص مثبت عليه نظام ويندوز .

يوجد لديك العديد من الخيارات و أشهرها البرنامج مفتوح المصدر ClamAV و هناك المغلق المصدر مثل BitDefender . و الآن سنتحدث بإذن الله عن BitDefender .

أولا نحتاج رخصة تتيح لنا إستخدام البرنامج لمدة ١٤٢ يوم و يمكننا طلبها مجانا من هنا :

```
http://www.bitdefender.com/site/Products/ScannerLicense
```

و تقوم بملئ البيانات المطلوبة كما في الصورة

United States - English | Contact Us | My BitDefender

bitdefender

enter query here ...

Company | Home/Home Office | Business | Partners | Downloads | Defense Center | Store

Products | Comparison Chart | Downloads | Support | Store

Home / Products /

Request free license

Please fill this registration, to receive a license for **BitDefender Antivirus Scanner for Unices**

First Name:* Mustafa

Last Name:* Refaey

E-mail:* @gmail.com

Phone:

Country: Egypt

All fields marked with * are necessary.

This license is free of charge and it can be retrieved from the BitDefender website after filling in a short form. It only allows the product to be used for personal purposes, with no commercial implications whatsoever, under the terms of the EULA. For example, under the Personal License, you are allowed to scan your personal laptop or desktop computer but **YOU ARE NOT ALLOWED TO USE THE PRODUCT IN A BUSINESS ENVIRONMENT SUCH AS AN OFFICE COMPUTER OR A COMPANY SERVER.**

☑ I agree

Send

ثم تضغط Send .. ستصلك بعدها رسالة على بريد الإلكتروني الذي وضعته سابقا بطلب الرخصة .. هذه الرسالة بها مفتاح الرخصة .

و الآن تقوم بتحميل البرنامج :

لأصحاب التوزيعات الديبيانية

```
http://download.bitdefender.com/SMB/Workstation_Security_and_Management/BitDefender_Antivirus_Scanner_for_Unices/Unix/Current/EN_FR_BR_RO/Linux/BitDefender-Antivirus-Scanner-7.6-4.linux-gcc4x.i586.deb.run
```

لأصحاب التوزيعات الريدهاتية

```
http://download.bitdefender.com/SMB/Workstation_Security_and_Management/BitDefender_Antivirus_Scanner_for_Unices/Unix/Current/EN_FR_BR_RO/Linux/BitDefender-Antivirus-Scanner-7.6-4.linux-gcc4x.i586.rpm.run
```

و للقائمة الكاملة من الحزم

```
http://download.bitdefender.com/SMB/Workstation_Security_and_Management/BitDefender_Antivirus_Scanner_for_Unices/Unix/Current/EN_FR_BR_RO/Linux/
```

و الآن نقوم بتثبيت البرنامج .. افتح الطرفية **Terminal** ثم نذهب للمسار الموجود به الحزمة و بحالتي أنا فهو كالتالي

```
cd /home/mustafa/Download
```

ثم نعطي الحزمة صلاحية التنفيذ

```
chmod +x BitDefender-Antivirus-Scanner-7.6-4.linux-gcc4x.i586.rpm.run
```

ثم نقوم بإدخال الأمر التالي بصلاحيات الجذر

```
./BitDefender-Antivirus-Scanner-7.6-4.linux-gcc4x.i586.rpm.run
```

نضغط على **Enter** حتى نصل لنهاية الرخصة ثم نكتب **accept** ثم نضغط **Enter** و سيسألك هل تريد واجهة البرنامج الرسومية فإذا كنت تريدها فاضغط **y** و إذا لم تردها فاضغط **n** ثم **Enter** و بهذا سيتم التثبيت و ينتهي بعد ثواني .

بعدها نقوم بفتح البرنامج عن طريق الأمر **bdgui**



نقوم بالضغط على **Set new key** ثم ندخل مفتاح الرخصة الذي حصلنا عليه بالرسالة منذ قليل ثم نضغط على **set** .
و للتعامل مع البرنامج فيلزم صلاحيات الجذر .. مثلا لو أردت تحديث البرنامج فنقوم بتنفيذ هذا الأمر بصلاحيات الجذر

```
bdscan --update
```

المحررين



الكتاب

ياسـر يكن
مينـا ألبـير
مصعب الزعبـي
مصطفـى رفاعـي
مؤيد السـعدي
فتحـي القدسـي
جمعـان الزهرانـي
أميـن روخ
أبـو عمـر
حمدي هشام مرشد

رئيس الإدارة

أ/ سـامر حداد

تصميم و إخراج

إبـراهيـم صقـر